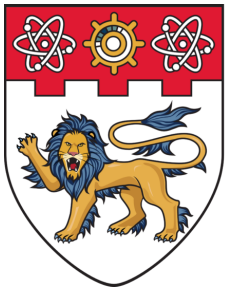

NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

TOWARDS ROBUST AND EFFECTIVE VISUAL LOCALIZATION

WANG SIJIE

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2025

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

31/05/2025

Date

Wang Sijie

WANG SIJIE

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

31/05/2025

Date

Prof. Tay Wee Peng

Authorship Attribution Statement

This thesis contains material from four paper(s) published in the following peer-reviewed journal(s) / from papers accepted at conferences in which I am listed as an author.

Parts of Chapters 2 and 3 are published as Sijie Wang, Qiyu Kang, Rui She, Wee Peng Tay, Andreas Hartmannsgruber, and Diego Navarro Navarro, "RobustLoc: Robust camera pose regression in challenging driving environments," in Proceedings of the AAAI Conference on Artificial Intelligence, 2023. The contribution of co-authors are as follows:

- Prof. Tay Wee Peng provided the initial project direction for using neural ordinary differential equations in localization.
- I designed detailed methods, wrote all experiment codes, and prepared the manuscript.
- Prof. Tay Wee Peng and Dr. Kang Qiyu revised the manuscript.

Parts of Chapters 2 and 4 are published as Sijie Wang, Qiyu Kang, Rui She, Wei Wang, Kai Zhao, Yang Song, and Wee Peng Tay, "HypLiLoc: Towards effective LiDAR pose regression with hyperbolic fusion," in Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference, 2023. The contribution of co-authors are as follows:

- Prof. Tay Wee Peng provided the initial project direction for using hyperbolic embedding in localization.
- I designed detailed methods, wrote all experiment codes, and prepared the manuscript.
- Prof. Tay Wee Peng and Dr. Kang Qiyu revised the manuscript.

Parts of Chapters 2 and 5 are published as Sijie Wang, Rui She, Qiyu Kang, Kai Zhao, Yang Song, and Wee Peng Tay, "PRFusion: Towards effective and robust multi-modal place recognition with image and point cloud fusion," IEEE Transactions on Intelligent Transportation Systems. The contribution of co-authors are as follows:

- Prof. Tay Wee Peng provided the initial project direction for using multi-modal sensors in place recognition.

- I designed detailed methods, wrote all experiment codes, and prepared the manuscript.
- Prof. Tay Wee Peng, Dr. Kang Qiyu, and Dr. She Rui revised the manuscript.

Parts of Chapters 2 and 6 are published as Sijie Wang, Rui She, Qiyu Kang, Xingchao Jian, Kai Zhao, Yang Song, and Wee Peng Tay, "DistilVPR: Cross-Modal knowledge distillation for visual place recognition," in Proceedings of the AAAI Conference on Artificial Intelligence, 2024. The contribution of co-authors are as follows:

- Prof. Tay Wee Peng provided the initial project direction for using knowledge distillation in localization.
- I designed detailed methods, wrote all experiment codes, and prepared the manuscript.
- Prof. Tay Wee Peng and Dr. She Rui revised the manuscript.

Parts of Chapters 2 and 7 are published as Sijie Wang, Rui She, Qiyu Kang, Siqi Li, Disheng Li, Tianyu Geng, Shangshu Yu, and Wee Peng Tay, "Multi-modal aerial-ground cross-view place recognition with neural ODEs," in Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference, 2025. The contribution of co-authors are as follows:

- Prof. Tay Wee Peng provided the initial project direction for using aerial database in localization.
- I designed detailed methods, wrote all experiment codes, and prepared the manuscript.
- Prof. Tay Wee Peng revised the manuscript.

31/05/2025

Date

WANG SIJIE

Acknowledgements

I am deeply grateful to the amazing individuals who have accompanied me throughout this Ph.D. journey.

First, I would thank my supervisor, Prof. Tay Wee Peng. He is really an all-around superman professor, who can excel in research, engineering, management, and social interaction. I have learnt a lot from him, and the four years are a fruitful journey to me.

Second, I would like to thank my parents and other family members, who have always supported me and stood by my side.

Third, I would thank all seniors and friends met in Prof. Tay's team and in Singapore. They are kind, intelligent, and friendly.

This thesis marks a significant milestone in my lifelong journey, where the 22-year learning journey comes to the end. I am proud that I am going to become the first Ph.D. in my Wang's family.

Summary

Visual localization is the task of determining an agent's location using visual sensors such as cameras and LiDARs. It plays a pivotal role in applications including autonomous navigation, robotics, and augmented reality, where precise positioning in complex environments is essential. However, existing approaches are not robust against environmental perturbations. Moreover, the fusion of multi-modal information from different sensors is not optimized for effective visual localization in both same-view and cross-view settings. Furthermore, there is a gap in efficiently transferring rich multi-modal information into single-modal pipelines, limiting the potential for lightweight and computationally efficient operation.

To address these challenges, we leverage neural ordinary differential equations to achieve robust visual localization under challenging conditions. The non-intersection property of neural ordinary differential equations also ensures the consistency between the scene feature and the actual geo-distance. Additionally, we propose to embed features into spaces of distinct curvatures to enhance their geometric representation ability. Furthermore, we incorporate learnable manifolds with flexible chart functions and tangent space metrics to effectively fuse features from different modalities. This supports more comprehensive sensor data interaction and thus improves multi-modal visual localization performance. In addition, we propose to combine self-agent and cross-agent relationships to enhance the effectiveness of localization knowledge transfer. The multi-relationship knowledge transfer increases the potential for lightweight single-modal localization with improved efficiency.

We evaluate these approaches across various visual localization scenarios. We also conduct extensive experiments on benchmark datasets to demonstrate the effectiveness of these approaches. These approaches hold significant potential for advancing autonomous systems and related technologies.

Contents

List of Figures

List of Tables

Symbols and Acronyms

Symbols

$\mathbb{R}^{d_1 \times d_2}$	The space of real matrices with d_1 rows and d_2 columns.
$\oplus$	The broadcast addition operation.
$\oplus_c$	The Mobius addition operation with the curvature parameter c.
Pooling$(\cdot)$	The global average pooling operation.
$\|\cdot\|$	The ℓ_2 norm.
$[N]$	The set of integers $\{1, 2, \ldots, N\}$.
Softmax$(\cdot)$	The softmax operation.
$\|\|(\cdot)$	The concatenating operation.
$\lfloor \cdot \rfloor$	The floor operation.
$\circ$	The function composition operation.
KNN$(\cdot)$	The K-nearest neighbor algorithm.
$\lvert\cdot\rvert$	The absolute value.
trace$(\cdot)$	The trace value.
$\exp_{\mathbf{z}}^{c}(\cdot)$	The exponential mapping operation at the base $\mathbf{z}$ to the hyperbolic space with curvature parameter c.

Acronyms

ODE	Ordinary Differental Equations
KD	Knowledge Distillation
CNN	Convolutional Neural Network
GNN	Graph Neural Network
FOV	Field of View
6-DoF	6 Degrees of Freedom
MLP	Multi-Layer Perceptron
BEV	Bird's-Eye-View

PDE	Partial Differential Equation
FC	Fully Connected
SOTA	State-of-the-Art
EKF	Extended Kalman Filter

Chapter 1

Introduction

This chapter gives an introduction to the task of visual localization. It provides an overview in terms of localization aims, types, modalities, and views. In addition, the chapter defines the scope of this research and presents the central questions addressed in the thesis. Finally, it offers a structural roadmap for the succeeding chapters.

1.1 About Localization

Localization is a fundamental task in wide fields. It aims at inferring the location of a system based on the provided query sensor data. Originally, Global Navigation Satellite System (GNSS) is the basic solution for localization by leveraging global satellites and stations [1]. In recent years, with the development of deep learning, there has been increasing interest in visual localization solutions that use more flexible mobile sensors such as cameras and LiDARs.

By extracting features from input sensor data, e.g., images from cameras and point clouds from LiDARs, deep neural networks are able to infer the location[2, 3]. Compared to traditional GNSS-based methods, visual localization offers several advantages: it can operate in GPS-denied environments such as indoors or urban canyons, adapts better to complex geometries and scene appearances, and supports both metric and semantic-level understanding of the environment.

However, visual localization also introduces new challenges including robustness and effectiveness. Visual localization systems are highly sensitive to input perturbations, such as changes in illumination, weather conditions, occlusions, or motion blur. These perturbations can significantly degrade the quality of extracted features, thereby reducing the reliability and accuracy of localization. Therefore, how to maintain the robustness of feature extraction would be a critical problem to localization. Moreover, effectively extracting informative and discriminative features from sensor inputs remains a key challenge. The quality of these features directly impacts the localization performance, especially in visually ambiguous or dynamically changing environments. Designing feature extractors that are both robust to noise and sensitive to task-relevant variations is therefore essential for achieving accurate and reliable localization.

In this thesis, we focus on learning-based visual localization approaches that integrate deep representations with robustness and effectiveness. Our goal is to develop localization models that are both robust and effective across diverse environments.

1.2 Localization Types

To achieve the localization of an agent, several types of pipelines can be utilized, each with different levels of precision and performance. These pipelines include approaches such as pose regression[2], place recognition[4], and point-wise registration[5–7]. Among these, pose regression and place recognition are generally used for coarse-level localization, providing meter-level accuracy while operating efficiently. These methods are suitable for tasks that require quick estimation of the agent's location, but they may lack the accuracy needed for more fine-grained localization.

On the other hand, point-wise registration serves the purpose of fine-grained localization, enabling the agent to achieve centimeter-level accuracy [6]. This approach is slower due to the increased computational demands.

A full localization solution typically combines both coarse and fine-grained techniques. Coarse-level localization, provided by pose regression or place recognition, is often employed as a quick initialization step to estimate the agent's position roughly. Once this initialization is complete, fine-grained localization, using point-wise registration, is employed to refine the pose further and deliver more accurate

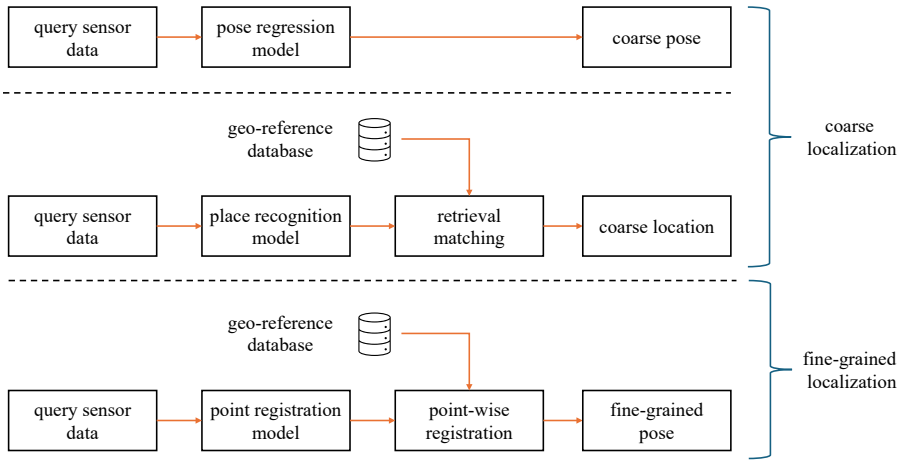

FIGURE 1.1: Localization type comparison. The above is pose regression, the middle is place recognition, and the below is point registration.

positioning. This combination ensures that the system benefits from both the speed of coarse localization and the precision of fine-grained techniques, allowing for a more robust and adaptable localization pipeline.

In this thesis, we focus on the coarse-level localization approaches, including pose regression and place recognition. The pipeline diagrams are shown in Fig. 1.1.

1.3 Localization Modalities

To achieve visual localization, different sensor options are available to produce visual data. These sensors generate data that can be easily visualized and typically contain rich information about the environment. This makes these sensors highly suitable for localization tasks.

Cameras are passive sensors that receive light and generate images containing dense visual information such as colors and textures. These images often provide rich and high-resolution details of the scene, allowing for fine-grained recognition of objects, landmarks, and environmental features[2, 4]. This dense and rich data makes cameras ideal for tasks that require detailed visual cues, though they can be sensitive to changes in lighting or weather conditions.

LiDARs, on the other hand, are active sensors that emit laser beams and calculate the distance to objects by measuring the laser flight time. This process allows LiDAR to generate sparse and accurate 3D point clouds that provide spatial information

about the surroundings. By capturing the geometry of the environment, LiDARs are particularly effective in perceiving the structure of objects and surfaces, making them valuable for localization[3, 4]. However, since LiDARs cast sparse lasers, the structural point cloud is produced in a sparse format. How to effectively handle the sparsity and extract informative features from point clouds remains a challenge for visual localization.

The fusion of different basic sensor modalities arises as another promising localization solution. For example, we can fuse images from cameras and point clouds from LiDARs, so that the colors, textures, as well as 3D structural information are all captured. This provides a much informative input into the visual localization neural network [8]. Therefore, how to effectively fuse different sensor modalities to build more representative scene-related features is critical for achieving accurate localization. On the other hand, the sensors that are additionally included would pose a burden on the system. They either result in high computational requirements or produce additional costs and weights for the system.

1.4 Localization Views

In the database-based localization type, visual localization approaches can be categorized into same-view localization or cross-view localization.

In same-view localization, query and database data are captured from the same view. For example, in ground-based localization scenarios, query and reference data are images captured on the ground, such as Google Street View data [9]. Similarly, in aerial photography, both the query and database data come from aerial sources, such as satellite [10] and unmanned aerial vehicle (UAV) imagery [11, 12]. This consistency in the same view simplifies the matching process, as data shares similar visual characteristics and scene perspectives.

On the other hand, cross-view localization involves query and database data captured from different views. A common example is that the query consists of ground-level images and the reference database is made up of aerial maps [13]. Cross-view localization is particularly valuable when same-view reference data is not readily available, such as in remote or wilderness areas where there is no pre-captured ground data. In these cases, aerial maps or satellite images may serve as the only

viable reference for localization. Cross-view localization poses unique challenges due to the inherent difference between different views.

1.5 Localization Applications

Relying only on flexible visual sensors, visual localization has been applied in a wide variety of fields. One of the largest fields would be autonomous driving which has gained significant interest. Visual localization enables autonomous vehicles to know where they are and thus can smoothly and safely operate on the road. In addition, smart robots can also be beneficial for visual localization. For example, they can have an accurate sense of location in both outdoor and indoor scenarios and can effectively conduct various tasks, e.g., delivery [14] and manufacture[15]. Moreover, visual localization is helpful in various challenging environments, such as forest [16], desert [17], and even other planets [18].

These diverse applications highlight the critical role of visual localization in enabling intelligent systems to perceive, understand, and interact with their environments in a robust and autonomous manner.

1.6 Cause of Lacking Robustness and Effectiveness

The cause of lacking robustness and effectiveness can be attributed to multiple aspects. First, the inherent limitations of sensor inputs fundamentally hinder robust localization. For instance, cameras, as passive sensors, are highly susceptible to environmental perturbations such as illumination changes and low-light conditions, which can drastically degrade the reliability of visual data.

Then, since visual localization heavily depends on feature extraction from neural networks, any perturbation in the input data inevitably leads to degraded feature quality, which in turn diminishes localization accuracy. Meanwhile, inappropriate design of neural networks may fail in exploiting information embedded in sensor inputs, leading to insufficient feature extraction and, consequently, ineffective visual perception.

Finally, visual localization networks rely on ground truth supervision to guide the optimization of network parameters. Without sufficiently informative and accurate guidance, networks cannot be trained reliably and are thus hindered from achieving effective localization.

1.7 Scopes and Research Questions

In this thesis, we address key questions to explore the task of coarse-level visual localization and introduce solutions for achieving robust and effective visual localization. An overview of the research gaps, core concepts, and motivations are presented in Table 1.1. The key components include graph neural networks (GNNs), ordinary differential equations (ODEs), manifolds, and knowledge distillation (KD).

In summary, our first contribution is a camera-based localization model designed to ensure robust performance under challenging conditions. Second, we develop a LiDAR-based point cloud localization model that enhances feature extraction from sparse point clouds. Third, we propose a multi-modal localization model that efficiently integrates information from both images and point clouds. Fourth, we introduce a KD pipeline to improve the localization performance of single-modal models. Lastly, we propose a cross-view localization model that leverages aerial maps to assist ground-level localization for mobile agents.

The contribution of this thesis is driven by the following research questions.

- Question 1: Can we enable camera-based localization with enhanced robustness, ensuring that the model can still perform effectively even in challenging driving conditions?

We introduce RobustLoc[19] in Chapter 3 to address this question. RobustLoc is built upon neural ODEs, and it allows for robust feature extraction under input perturbations. Furthermore, RobustLoc leverages neighboring image frames to aggregate additional information, thereby mitigating the information loss caused by sensor noise.

Our Work	Research Gap	Basic Idea	Motivation
RobustLoc	Current camera localization methods struggle under challenging conditions with environmental perturbations and significant sensor noise.	Leverage ODEs to enhance features and GNNs to incorporate neighboring information. Both can contribute to improved robustness against perturbations.	Neural ODEs can naturally handle input perturbations, while GNNs effectively aggregate neighbor data and mitigate the information loss caused by perturbations.
HypLiLoc	Existing LiDAR localization approaches fail to utilize geometric information from sparse point clouds, resulting in inadequate feature extraction.	Use both hyperbolic and Euclidean spaces to embed geometric features. Integrate point-based and projection-based backbones simultaneously.	Hyperbolic and Euclidean spaces allow for capturing geometric features with different curvatures, while point and projection backbones handle feature aggregation in different neighborhood formats.
PRFusion	Multi-modal localization methods often overlook integrating global summarized information and local fine-grained details for effective feature fusion.	Apply a learnable manifold metric for feature attention computing on flexible manifolds and use camera-LiDAR extrinsic calibrations for the pixel-level feature correspondence.	Manifolds with learnable metrics provide a more adaptable space for aligning features from different modalities, while extrinsic calibration supports finer-grained local feature fusion.
DistilVPR	Existing KD methods do not fully consider the intrinsic properties of matching-based localization across teachers and students.	Leverage both cross-agent and self-agent relationships. Incorporates geometric embeddings with different curvatures.	Various agent relationships help better explore query-database matches, and diverse curvature spaces offer a more flexible platform for exploring geometric relationships.
AGPlace	Existing multi-modal localization pipelines do not leverage a surrogate branch to align features from different views.	Utilize the non-intersection property of ODEs to create a surrogate space for feature interaction.	ODEs facilitate more efficient feature extraction that can be aligned with the geo-distance, while the surrogate manifold space bridges feature representations across different views.

TABLE 1.1: Summary of the research gaps, basic ideas, and motivations behind our proposed works.

- Question 2: Can we delve deeper into the properties of point cloud data to extract more hidden and informative features for more effective LiDAR-based localization?

We introduce HypLiLoc[20] in Chapter 4 to address this question. HypLiLoc takes advantage of space feature embeddings, including flat Euclidean space and

curved hyperbolic space. These multi-space embeddings enable the model to better understand point clouds. Additionally, HypLiLoc includes both point-wise and projection-based backbones, allowing for diverse neighborhood construction to support feature aggregation.

- Question 3: Can we effectively fuse different input modalities, incorporating both global summarized information and local fine-grained details, to build more powerful scene descriptors for robust and effective multi-modal localization?

We introduce PRFusion++ in Chapter 5 to address this question. PRFusion++ uses manifold metrics to support flexible attention that facilitates feature interaction between different modal spaces. Additionally, PRFusion++ leverages camera-LiDAR extrinsic calibrations to support fine-grained and pixel-level feature correspondence. This leads to more effective local-level feature fusion.

- Question 4: Can we transfer the advanced localization ability of multi-modal models into smaller and single-modal models, so that lightweight models operating on resource-constrained platforms can benefit from effective teacher guidance and achieve better performance?

We introduce DistilVPR[21] in Chapter 6 to address this question. DistilVPR facilitates knowledge transfer by considering relationships between teachers and students in both cross-agent and self-agent formats, which allows for more effective transfer of matching-based knowledge. Moreover, DistilVPR utilizes various curvature manifold embeddings to enhance the exploration of geometric features.

- Question 5: Can we design a more seamless network architecture to better align localization features from different views, thereby achieving more accurate cross-view localization?

We introduce AGPlace in Chapter 7 to address this question. AGPlace includes a surrogate manifold branch to align features from different views and modalities. It also incorporates neural ODEs to ensure that the constructed scene descriptors remain consistent with the actual geometric locations.

1.8 Thesis Organization

This thesis is structured as follows:

In Chapter 2, we introduce related works and fundamental concepts that form the foundation for the thesis. This chapter covers key theoretical preliminaries, the main techniques used in the literature, and essential background knowledge.

The following five chapters introduce our proposed solutions. Each chapter presents a specific approach, along with the experiments conducted on various datasets to validate the effectiveness of the approach.

In Chapter 3, we introduce RobustLoc, a camera pose regression model designed for localization tasks. Built upon neural ODEs, RobustLoc incorporates adjacent image frames to enhance resilience against input perturbations by enabling robust feature extraction.

In Chapter 4, we present HypLiLoc, a LiDAR-based pose regression model that benefits from spatial feature embeddings across both Euclidean and hyperbolic spaces. These multi-space representations improve the model's ability to interpret point clouds. HypLiLoc further integrates both point-wise and projection-based backbones, supporting diverse neighborhood constructions for effective feature aggregation.

In Chapter 5, we propose PRFusion and its extended variant PRFusion++, two multi-modal place recognition models that employ manifold-aware attention mechanisms to facilitate interaction across different modality spaces. PRFusion++ additionally utilizes precise camera-LiDAR extrinsic calibration to achieve fine-grained, pixel-level feature alignment, leading to enhanced local fusion performance.

In Chapter 6, we introduce DistilVPR, a cross-modal KD framework tailored for place recognition. It promotes efficient knowledge transfer by capturing both cross-agent and self-agent relationships between teacher and student models. DistilVPR also leverages manifold embeddings with varying curvature to better explore geometric structures.

In Chapter 7, we propose AGPlace, a model for aerial-ground cross-view place recognition. It incorporates a surrogate manifold branch to align multi-view and multi-modal features.

Finally, in Chapter 8, we revisit the proposed visual localization approaches in the thesis and suggest potential directions for future research that advances the field of visual localization.

Chapter 2

Background

This chapter provides a comprehensive review of the background of visual localization solutions. We cover the task introduction, existing approaches, and preliminaries that inform the solutions proposed in this thesis.

2.1 Task Introduction

We first introduce the basic background of the two visual localization types, including database-free pose regression and database-relied place recognition.

2.1.1 Pose Regression

In the pose regression pipeline, the implicitly given information during inference is the coordinate frame, where the regression model is required to estimate the sensor's 6-degree-of-freedom (6-DoF) pose with respect to this coordinate frame. The inputs to the regression model include images captured by cameras and point clouds produced by LiDARs.

Given a query sensor data $\mathbf{Q}$ (such as an image and a point cloud), the goal of pose regression network $f_{\text{pose}} : \mathbf{Q} \mapsto (\mathbf{d}, \mathbf{r})$ is to predict the 6-DoF sensor translation and rotation $\text{Pose} = (\mathbf{d}, \mathbf{r})$ with respect to a coordinate frame.

The model is trained to minimize a loss function that jointly considers both translation and rotation. A common loss formulation is:

$$\mathcal{L} = \alpha\|\mathbf{d} - \mathbf{d}^*\| + \beta\|\mathbf{r} - \mathbf{r}^*\| \tag{2.1}$$

where $\mathbf{d}^*$ and $\mathbf{r}^*$ denote the ground truth translation and rotation vectors. $\|\cdot\|$ denotes the ℓ_2 norm. The parameters α and β are scalar weights used to balance the importance of translation and rotation errors in the overall loss.

The objective of the model is to minimize this loss function and ensure that the predicted pose can closely match the ground truth pose in terms of both position and orientation.

2.1.2 Place Recognition

In contrast to pose regression, place recognition involves using an informative reference database that consists of geo-tagged sensor data. The task is to find the closest match in the reference database, such that the query scene or place can be recognized and its rough location can be estimated from the corresponding geo-tags.

Given a query input $\mathbf{Q}$ and a reference database $\{(\mathrm{Ref}_i, \mathbf{d}_i)\}_{i=1}^N$ that consists of the reference data Ref_i associated with the geo-tagged position $\mathbf{d}_i$, the goal of place recognition is to retrieve the reference data $\mathrm{Ref}_{\mathrm{top}}$ that most closely matches the query $\mathbf{Q}$.

Place recognition is usually formulated as a retrieval task, where a similarity score $\mathrm{Sim}(\mathbf{Q}, \mathrm{Ref}_i)$ is computed between the query and each reference entry. Thus, the objective of place recognition can be formulated as:

$$(\mathrm{Ref}_{\mathrm{top}}, \mathbf{d}_{\mathrm{top}}) = \underset{\mathrm{Ref}_i, \mathbf{d}_i}{\operatorname{argmax}} \mathrm{Sim}(\mathbf{Q}, \mathrm{Ref}_i) \tag{2.2}$$

where $\mathrm{Ref}_{\mathrm{top}}$ is the top-matched reference data.

To perform place recognition, both the query $\mathbf{Q}$ and the reference data Ref_i are embedded into a common feature space using deep neural networks $f_{\text{place}} : \mathbf{Q}, \text{Ref}_i \mapsto \mathbf{e}_{\mathbf{Q}}, \mathbf{e}_{\text{Ref}_i}$, where $\mathbf{e}_{\mathbf{Q}}, \mathbf{e}_{\text{Ref}_i}$ are the embeddings of the query and reference data. Therefore in the place recognition pipeline, the similarity score $\text{Sim}(\mathbf{Q}, \text{Ref}_i)$ is typically computed on the embeddings instead of the original sensor data, which can be formulated as:

$$\text{Sim}(\mathbf{Q}, \text{Ref}_i) = \frac{\mathbf{e}_{\mathbf{Q}} \cdot \mathbf{e}_{\text{Ref}_i}}{\|\mathbf{e}_{\mathbf{Q}}\| \|\mathbf{e}_{\text{Ref}_i}\|}. \tag{2.3}$$

During training, a loss function is used to optimize the neural network by maximizing the similarity between a query and its corresponding positive reference embedding while minimizing the similarity with irrelevant negative embedding. A commonly used loss for place recognition is the triplet loss:

$$\mathcal{L} = \max\big(\left\|\mathbf{e}^{\text{anchor}} - \mathbf{e}^{\text{positive}}\right\| - \left\|\mathbf{e}^{\text{anchor}} - \mathbf{e}^{\text{negative}}\right\| + m, 0\big), \tag{2.4}$$

where m is the margin hyperparameter. $\mathbf{e}^{\text{anchor}}, \mathbf{e}^{\text{positive}}, \mathbf{e}^{\text{negative}}$ are the embeddings of an anchor query, a positive reference, and a negative reference, respectively.

Once the top-matched reference data Ref_{top} is retrieved, the corresponding location $\mathbf{d}_{\text{top}}$ is used to estimate the location of the query. This provides a coarse localization of the query scene.

2.2 Related Works

In this section, we provide a literature summary of the related works in visual localization.

2.2.1 Camera Pose Regression

Given the query images, camera pose regression models directly regress the camera poses of these images without the need for a database. Thus, it does not depend on

the scale of the database, which is definitely a born gift compared with database approaches.

PoseNet [2] and GeoPoseNet [22] pioneer to directly regress the pose of the sensor data. They construct pose regression models that consist of a feature extraction backbone and a pose regression head as shown in Fig. 2.1. There are various backbones used for visual feature extraction, such as convolutional neural networks (CNNs) (e.g., ResNet[23], ConvNeXt[24]) and Transformers (e.g., ViT[25] and Swin[26]). These backbones are usually pre-trained on the large-scale ImageNet dataset[27] such that they have sufficient ability to recognize general image patterns for other tasks. GeoPoseNet simultaneously learns locations and orientations by introducing balancing weights to account for the scales of these two variables. This method ensures more accurate pose estimation.

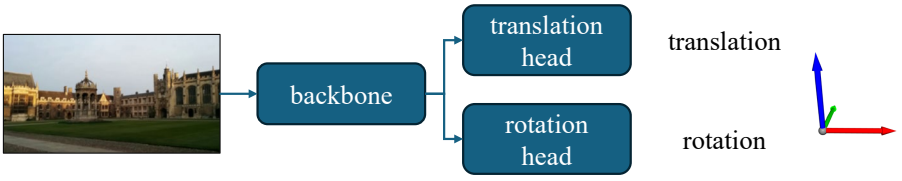

FIGURE 2.1: Overview of a pose regression model.

MapNet [28] integrates inexpensive and widely available sensory inputs, such as the odometry trajectory, with image data for enhanced camera localization. It incorporates geometric constraints by formulating them as loss terms during training, which is formulated as:

$$\mathcal{L} = \sum_{i,j} h_{\text{MapNet}}(\text{Pose}_{i,j}, \text{Pose}^*_{i,j}), \tag{2.5}$$

where $\text{Pose}_{i,j}, \text{Pose}^*_{i,j}$ are the predicted and ground truth relative poses between i and j frames respectively. h_{MapNet} is the loss function to measure the prediction and ground truth pose, which is defined as:

$$h_{\text{MapNet}}(\text{Pose}, \text{Pose}^*) = \|\mathbf{d} - \mathbf{d}^*\| e^{-\alpha} + \alpha + \|\mathbf{r} - \mathbf{r}^*\| e^{-\beta} + \beta, \tag{2.6}$$

with balancing parameters α, β.

AD-PoseNet and AD-MapNet [29] leverage semantic masks to exclude dynamic areas from images. The key innovation in these models is the prior-guided dropout

module, which guides the network to ignore foreground objects during both training and inference Fig. 2.2. Additionally, the dropout module enables the pose regressor to generate multiple hypotheses, facilitating the quantification of pose estimate uncertainty. This uncertainty is then exploited in an uncertainty-aware pose graph optimization, which significantly improves the robustness of the overall pose estimation process.

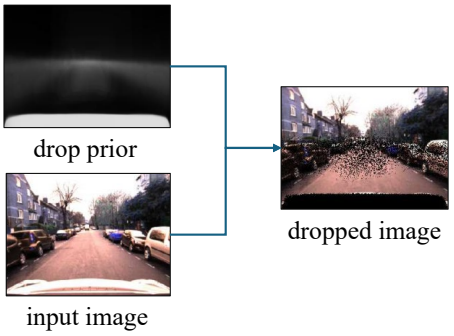

FIGURE 2.2: Visualization of the prior-guided dropout.

AtLoc [30] leverages global attention mechanisms to guide the network towards more geometrically robust objects and features, enhancing the spatial consistency of the scene representation. Additionally, it incorporates temporal constraints between image pairs to improve the consistency and accuracy of the learned features across sequences.

Coordinet [31] employs CoordConv [32] and weighted average pooling [33] techniques to effectively capture spatial relationships. Additionally, it integrates an uncertainty estimation module for the pose. The network jointly learns both the pose and its uncertainty using a unified loss function. During inference, these poses are refined by the Extended Kalman Filter (EKF).

2.2.2 LiDAR Pose Regression

LiDAR sensors actively emit beams to estimate sparse depth information in the environment. They are more robust against illumination changes compared to camerasFig. 2.3. This resilience has made LiDAR a critical component in various applications[34–36].

FIGURE 2.3: Visualization of the point cloud.

PointLoc [3] pioneers to leverage LiDAR point clouds for pose regression by utilizing PointNet++ [37] followed by self-attention to extract point cloud features. PointNet++ is a commonly used point-wise backbone for point cloud feature extraction. In each layer, centroids are selected from the input point cloud. These centroids will aggregate the neighborhood information from other points and then be used as the input into the next layer Fig. 2.4.

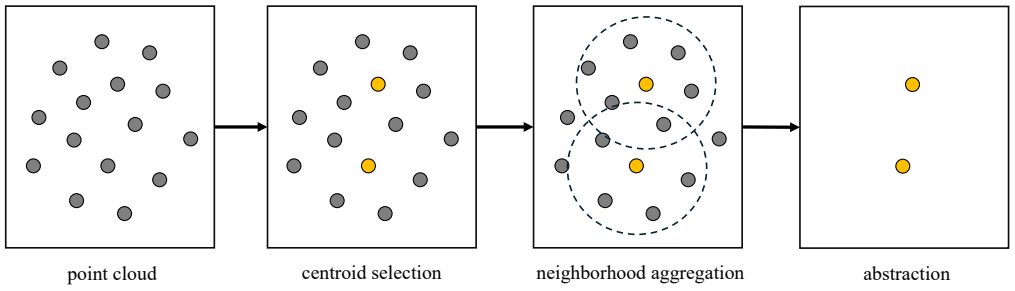

FIGURE 2.4: Overview of a layer in PointNet++.

Later, this paper [38] proposes a universal feature encoding approach to handle various scenes efficiently. This method enables high efficiency and data privacy. Additionally, it introduces a memory-aware regressor that adjusts the hidden unit to control the memorization capacity, which allows for the increased upper bound of the network capacity.

SGLoc[39] effectively encodes the scene coordinates which are preserved in the scene geometry, rather than relying solely on a global LiDAR pose. This method of additional scene coordinate encoding results in significant performance improvements. In addition, SGLoc integrates a tri-scale spatial feature aggregation module and a geometric constraint loss. These components work together to effectively capture and utilize scene geometry, enhancing overall localization accuracy.

2.2.3 Image Place Recognition

Different from pose regression approaches, place recognition solutions assume there exists a reference database to store reference scene data or extracted reference scene descriptors Fig. 2.5.

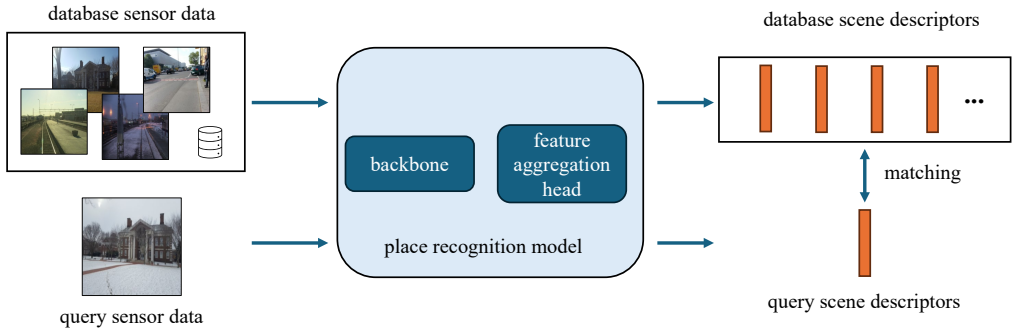

FIGURE 2.5: Pipeline of the place recognition model.

Traditionally, hand-crafted models are used to extract scene descriptors from images, such as BoW [40] and VLAD [41]. BoW represents an image by quantizing local features into a histogram of visual words. It involves extracting local descriptors and clustering them into a visual vocabulary. Each image is then represented by a histogram of the frequency of visual words. VLAD improves BoW by aggregating local descriptors more effectively. Instead of merely counting visual words, VLAD aggregates local feature descriptors around the cluster centers and encodes them into a compact vector. This method captures more detailed information about the descriptors' distribution. However, these methods heavily rely on prior information, which may not perform well in challenging environments. To address the limitations of traditional methods, advances in deep learning have inspired learning-based place recognition models. NetVLAD [42] pioneers the combination of the traditional VLAD descriptor with a CNN to construct a learnable feature aggregation layer. Its success has paved the way for many image-based place recognition models.

Building on the aggregation concept from NetVLAD, there are a series of works [43–45] extending the idea further. For example, GeM [43] generalizes max and average pooling operations into a flexible pooling layer with a learnable parameter. This parameter allows for adaptive channel weighting, enhancing the representation

of important features. The GeM layer is formulated as:

$$f_{\mathrm{GeM},p}(\mathbf{F}) = \left(\frac{1}{|\mathbf{F}|} \sum_{\mathbf{F}_i \in \mathbf{F}} \mathbf{F}_i^p \right)^{1/p}, \tag{2.7}$$

where p is the learnable parameter, $\mathbf{F}$ is the feature map, and $\mathbf{F}_i$ is the i-th feature point in the feature map $\mathbf{F}$.

Building on GeM, MixVPR [45] advances feature aggregation strategies by incorporating multi-layer perceptrons (MLPs) to process holistic feature maps. MixVPR utilizes MLPs to perform both channel-wise and spatial-wise feature mixing. It improves the scene representation ability by combining features across different dimensions and scales.

In contrast to these aggregation-focused solutions, other works design holistic feature extraction pipelines [46–50]. For example, PatchNetVLAD [46, 47] advances both local and global descriptor methods by deriving patch-level features from NetVLAD residuals. R2Former [50] integrates retrieval and re-ranking procedures to create more cohesive scene descriptors. By balancing global summarization with local detail preservation, R2Former effectively combines both global context and fine-grained local information.

However, one drawback of utilizing images is their susceptibility to environmental changes, such as fluctuating illumination and varying weather conditions.

2.2.4 Point Cloud Place Recognition

In contrast to images, point clouds produced by LiDARs demonstrate greater robustness against environmental perturbations [20, 39]. Since LiDARs actively emit laser beams, they are less susceptible to illumination perturbations.

PointNetVLAD [4] marks a significant advancement by utilizing point clouds rather than images for place recognition. It extracts point cloud features using PointNet [37, 51], which are then processed by a NetVLAD layer to generate the final global descriptor of the scene. This approach has inspired a series of 3D-based models aimed at enhancing point cloud place recognition.

Following the success of PointNetVLAD, several works have leveraged PointNet-series for point-based feature extraction [52–55]. These methods are built on the foundational PointNet architecture [37, 51]. For example, EPC-Net utilizes a spatial adjacency matrix and proxy points to streamline the original edge convolution, thereby reducing memory consumption. Additionally, it introduces a group layer that decomposes high-dimensional vectors into several low-dimensional groups. This approach not only decreases the number of network parameters but also preserves the discriminative information in the feature vectors.

In addition to point-based methods, voxel-based (cube-based) feature extraction has gained attention [56–58]. For instance, MinkLoc3D [56] pioneers the use of sparse 3D convolutional layers [59] to extract voxelized point cloud features Fig. 2.6. Voxel convolution operates over these voxel grids, sliding a 3D kernel across the input voxels to extract spatial features. Similar to 2D convolution in images, voxel convolution captures spatial relationships between neighboring voxels, enabling the model to learn geometric patterns and structures from 3D data.

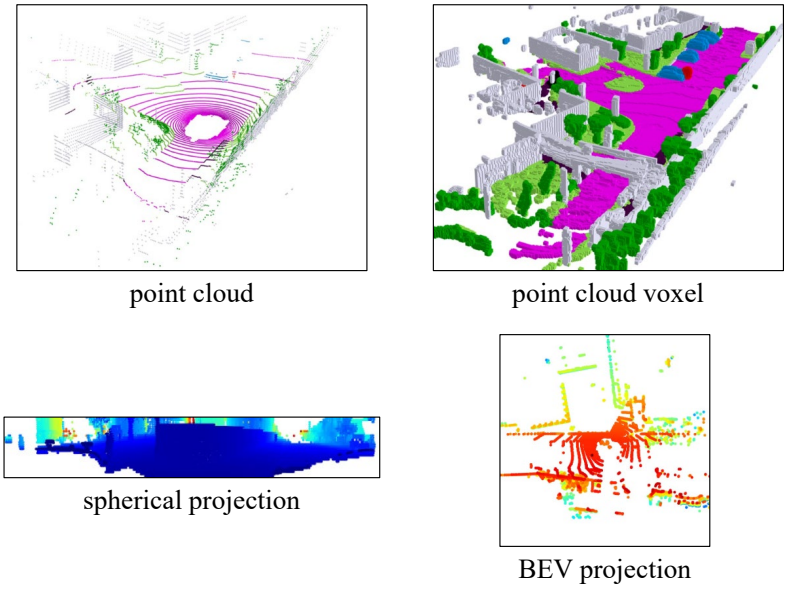

FIGURE 2.6: Comparison between point clouds and voxels.

Another direction involves projecting the 3D point cloud into 2D formats, employing spherical projection [60–62] and bird's-eye-view (BEV) methods [63]. These projection techniques offer alternative ways to interpret 3D data by leveraging information available in 2D representations. Among these methods, BEVPlace [63] transforms raw point clouds into a BEV format, enabling the application of image-like feature extraction techniques. The visualization of different point cloud

projection methods is shown in Fig. 2.6. It employs group convolution layers to capture rotation-invariant local features.

2.2.5 Multi-Modal Place Recognition

Recent research has demonstrated that incorporating multiple modalities can improve performance compared to using a single one. This has led to a growing interest in multi-modal learning using both images and point clouds for place recognition. The process of multi-modal place recognition typically employs a two-branch design, where images and point clouds are processed through their respective backbones for basic feature extraction Fig. 2.7. Then the extracted multi-modal features are fused with fusion layers.

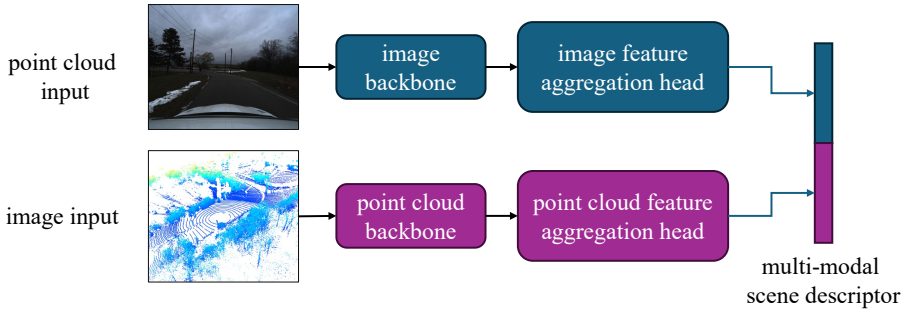

FIGURE 2.7: Multi-modal scene descriptor construction.

Some multi-modal place recognition works [8, 64–66] focus on fusing multi-modal features at the global level, often neglecting fine-grained local information. For instance, MinkLoc++[8] and AdaFusion[66] perform multi-modal feature fusion only on globally-pooled descriptors.

To address this limitation, there are other solutions [18, 67] that attempt to incorporate fine-grained level information by fully integrating the entire feature maps extracted by the two branches. Specifically, LCPR [67] leverages self-attention to fuse image features and projected spherical point cloud features. UMF [18] employs both self-attention and cross-attention to capture patterns. It also leverages MAE-based[68] network pre-training pipeline to enhance the feature extraction ability of 2D and 3D backbones. During pre-training, the image pixels and point cloud voxels are masked randomly, and the backbones are forced to reconstruct the masked area. A visualization of the image-based MAE is shown in Fig. 2.8.

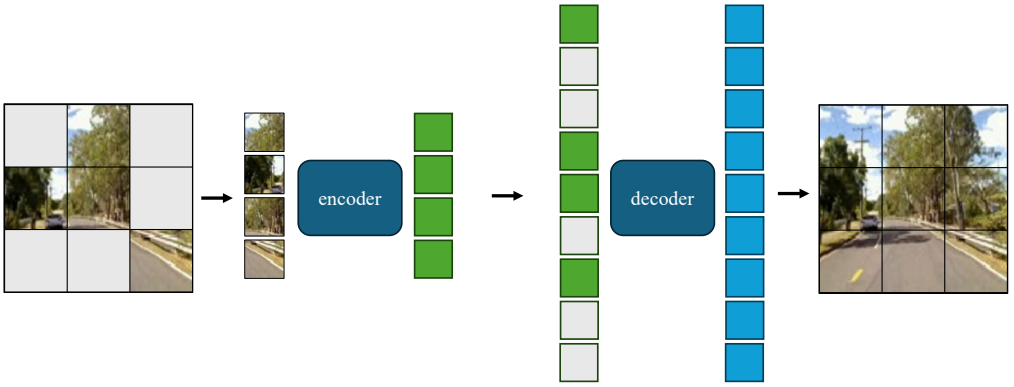

FIGURE 2.8: Visualization of MAE. The image is randomly masked, and the goal of MAE model is to reconstruct the masked area.

Despite these advancements, existing multi-modal place recognition models still rely on simplistic fusion techniques, such as concatenation [8] and basic attention mechanisms [18, 66, 67]. Moreover, they do not exploit camera-LiDAR extrinsic parameters to facilitate effective local fusion. There remains significant potential to enhance multi-modal feature interaction and improve overall performance.

In contrast, utilizing aerial maps as the database offers a potential solution to address the aforementioned challenges, yet it remains an under-explored area [69].

There are several advantages to using aerial maps: (1) They are readily available and include various types such as satellite images and road maps. (2) Aerial maps inherently capture a larger field-of-view (FOV) compared to ground sensor data, allowing place recognition models to operate on larger scales. (3) Collecting aerial data is more efficient than ground data, especially in challenging terrains like forests and mountains. These factors motivate us to investigate aerial-ground multi-modal place recognition that could benefit ground agent localization.

2.2.6 Knowledge Distillation

KD has emerged as a pivotal technique in model compression and multi-modal learning, enabling the transfer of knowledge from complex teacher models to compact or cross-modal student models as shown in Fig. 2.9. Vanilla KD [70] first introduces the concept of KD to compress the knowledge from larger teacher models to smaller student models. Given a teacher model $f_{\text{tea}}(\cdot)$ and a student model $f_{\text{stu}}(\cdot)$, the

KD loss can be formulated as:

$$\mathcal{L}(\mathbf{Q}) = \ell(f_{\text{tea}}(\mathbf{Q}), f_{\text{stu}}(\mathbf{Q})), \tag{2.8}$$

where $\ell(\cdot, \cdot)$ is the loss function. The KD loss is applied to the student model to optimize the model parameters. RKD [71] emphasizes the self-agent relationships of both teacher and student outputs. This approach serves as an implicit distillation solution, facilitating the effective transfer of the teacher's knowledge to the student model. An RKD loss can be formulated as:

$$\mathcal{L}(\mathbf{Q}_i, \mathbf{Q}_j) = \ell(d(f_{\text{tea}}(\mathbf{Q}_i), f_{\text{tea}}(\mathbf{Q}_j)), d(f_{\text{stu}}(\mathbf{Q}_i), f_{\text{stu}}(\mathbf{Q}_j))), \tag{2.9}$$

where $d(\cdot, \cdot)$ is the distance/relationship measurement function.

AFD [72] employs an attention-based meta-network to acquire relative similarity among features and then employs the similarity to regulate the intensity of distillation for all feasible pairs. MKD [73] conducts prediction alignment at three different levels simultaneously, which include the instance, batch, and class levels.

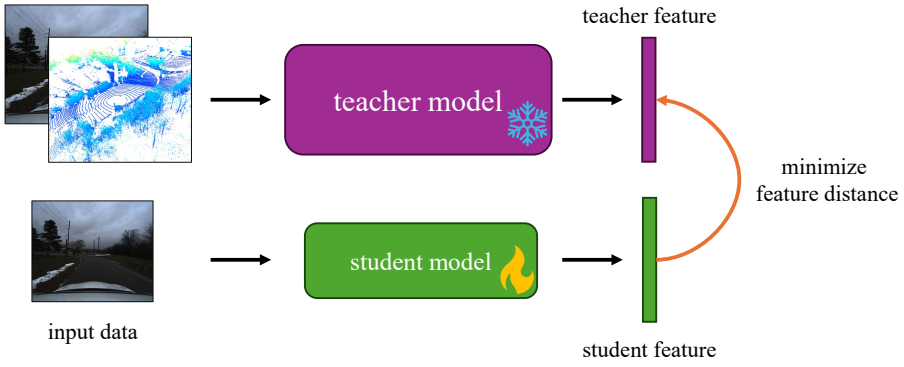

FIGURE 2.9: Cross-modal KD pipeline. The stronger teacher model takes as input multi-modal data, while the weaker student model takes as input single-modal data.

There are also some other distillation works focusing on various tasks. 2DPass [34] employs an innovative approach to enhance the point cloud semantic information extraction by distilling knowledge from images. LSD-Net [74] leverages dual distillation to transfer teacher patterns into students for lightweight place recognition. LiDAR2Map [35] presents an online camera-to-LiDAR distillation scheme to facilitate semantic information from images to point clouds for semantic map segmentation.

2.3 Preliminaries

2.3.1 Neural Differential Equations

ODEs serve as a fundamental tool for describing dynamical systems. They express how a system state evolves over time and capture relationships between the system variables and the rates of change:

$$\frac{d\gamma(t)}{dt} = f(\gamma(t), t), \tag{2.10}$$

where $\gamma(t)$ denotes the ODE state and f denotes the function to describe the ODE state. The paper [75] first proposes trainable neural ODEs by parameterizing the continuous dynamics of hidden units. The hidden state of the ODE network is modeled as:

$$\frac{d\gamma(t)}{dt} = f_\theta(\gamma(t), t), \tag{2.11}$$

where f_θ is the trainable network parameterized by weights θ. Recent studies [76–81] have demonstrated that neural ODEs are intrinsically more robust against input perturbations compared to vanilla CNNs.

Theorem 1. (Non-intersection of ODE solutions.)[76, 82] Given an ODE $\frac{d\gamma(t)}{dt} = f(\gamma(t), t)$, where f is continuous in t and globally Lipschitz continuous in γ. Let $\gamma_1(t)$ and $\gamma_2(t)$ be two solutions of the ODE. If there exits initial conditions $\gamma_1(0) \neq \gamma_1(0)$, then it holds that $\gamma_1(t) \neq \gamma_2(t)$ for all $t \in [0, \infty)$.

The non-intersection of ODE solutions presented in Theorem 1 indicates that different ODE inputs guarantee different ODE outputs, which can be used to construct more distinguished scene descriptors that align with the actual geo-distances.

Furthermore, the non-intersection also indicates the inner robustness of ODEs. Let $\gamma_1(0) < \gamma_2(0) < \gamma_3(0)$ are three different initial values. Then according to Theorem 1, the three solutions maintain the order $\gamma_1(t) < \gamma_2(t) < \gamma_3(t)$, i.e., $\gamma_2(t)$ is bounded. This demonstrates the inner robustness of ODEs.

In addition, neural partial differential equations (PDEs) [83, 84] have been proposed and applied to GNNs, where the diffusion process is modeled on the graph. Furthermore, the stability of the heat semigroup and the heat kernel under perturbations of the Laplace operator (i.e., local perturbation of the manifold) is studied [78]. Neural diffusion equations are also applied to image patch matching[85–87] and point cloud registration[88, 89].

2.3.2 Manifolds

The concept of a manifold serves as a generalization of surfaces in higher dimensions, extending the notion of well-behaved geometrical structures. An n-dimensional manifold $\mathcal{M}$ is a topological space that locally resembles the topological space $\mathbb{R}^n$ near each point $\mathbf{p} \in \mathcal{M}$. For each point $\mathbf{p}$, it is possible to establish a homeomorphism between a neighborhood of $\mathbf{p}$ and $\mathbb{R}^n$.

The tangent space $T_{\mathbf{p}}\mathcal{M}$ at a point $\mathbf{p}$ on $\mathcal{M}$ can be visualized as a n-dimensional hyperplane in $\mathbb{R}^{n+1}$ that provides the best approximation of $\mathcal{M}$ in the vicinity of $\mathbf{p}$. Alternatively, $T_{\mathbf{p}}\mathcal{M}$ is the space that encompasses all the possible directions of curves on $\mathcal{M}$ passing through $\mathbf{p}$. The elements residing within $T_{\mathbf{p}}\mathcal{M}$ are tangent vectors, and the collection of all tangent spaces forms the tangent bundle. Essentially, the tangent space $T_{\mathbf{p}}\mathcal{M}$ characterizes the local linear approximation of $\mathcal{M}$ near the point $\mathbf{p}$. It captures the intrinsic geometry of $\mathcal{M}$.

A metric tensor, also known as a metric, is an additional structure defined on a manifold $\mathcal{M}$ that enables the measurement of distances and angles, similar to how the inner product in Euclidean space allows for the definition of distances and angles. Specifically, at each point $\mathbf{p}$ on $\mathcal{M}$, a metric tensor is a bilinear form defined on the tangent space at $\mathbf{p}$. This bilinear form takes pairs of tangent vectors and assigns real numbers as $\mathbf{g}(\mathbf{p}) : T_{\mathbf{p}}\mathcal{M} \times T_{\mathbf{p}}\mathcal{M} \to \mathbb{R}$. By smoothly varying across $\mathcal{M}$, the metric tensor provides a consistent way to measure distances and angles throughout the manifold.

2.3.3 Curvature Spaces

Different feature manifolds can be categorized based on their curvature [90]. The Euclidean space represents the most prevalent manifold with zero curvature, while the spherical manifold exhibits positive curvature, and the hyperbolic manifold has negative curvature Fig. 2.10. By using multiple manifolds, we can facilitate features to possess more comprehensive embedding relationships by leveraging distinct geodesic distances.

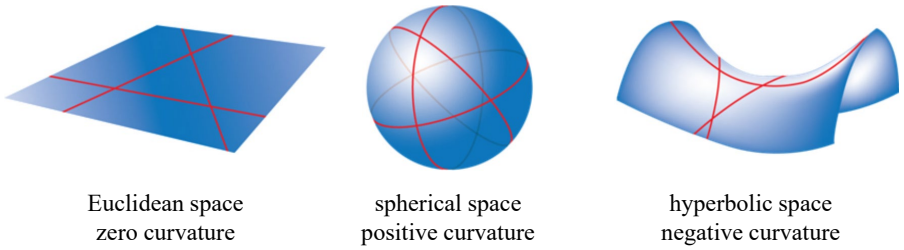

FIGURE 2.10: Visualization of Euclidean, spherical, and hyperbolic spaces.

Euclidean Space. Euclidean space serves as a prominent example of a flat manifold, exhibiting zero curvature across all points. Within Euclidean space, the calculation of the geodesic distance between any two points is given by the conventional Euclidean distance. The distance d_{euc} is the straight-line distance between two points in a Cartesian coordinate system given by:

$$d_{\text{euc}}(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\| \,. \tag{2.12}$$

Spherical Space. In contrast to Euclidean space, the spherical manifold displays a distinct characteristic by possessing a constant positive curvature. The geodesic distance between two points is calculated based on the angular separation between the points and the radius of the sphere. The cosine distance can be viewed as the spherical distance on a unit sphere:

$$d_{\cos}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \, \|\mathbf{y}\|}. \tag{2.13}$$

Hyperbolic Space. In Riemannian geometry, the hyperbolic space is defined as the Riemannian manifold with constant negative curvature. The Poincaré ball is the most common conformal model of hyperbolic geometry. It has been used to embed features in various tasks [91–93]. The n-dimensional Poincaré ball is defined

on $\mathbb{D}_c^n = \{\mathbf{p} \in \mathbb{R}^n : c\|\mathbf{p}\| < 1\}$ with curvature $-c^2$. The Poincaré ball is equipped with a metric tensor $\mathbf{g} = \lambda_c^2 \mathbf{E}^n$, where $\lambda_c = \frac{2}{1-c\|\mathbf{p}\|^2}$ is the conformal factor and $\mathbf{E}^n$ is the identity matrix.

Given a pair $\mathbf{p}, \mathbf{q} \in \mathbb{D}_c^n$, the mobius addition $\oplus_c$ is defined as:

$$\mathbf{p} \oplus_c \mathbf{q} = \frac{\left(1 + 2c\langle \mathbf{p}, \mathbf{q}\rangle + c\|\mathbf{q}\|^2\right)\mathbf{p} + \left(1 - c\|\mathbf{p}\|^2\right)\mathbf{q}}{1 + 2c\langle \mathbf{p}, \mathbf{q}\rangle + c^2\|\mathbf{p}\|^2\|\mathbf{q}\|^2}. \tag{2.14}$$

For a fixed base point $\mathbf{z} \in \mathbb{D}_c^n$, the exponential mapping function $\exp_{\mathbf{z}}^c : \mathbb{R}^n \to \mathbb{D}_c^n$ maps points from the tangent Euclidean space to the hyperbolic space:

$$\exp_{\mathbf{z}}^c(\mathbf{v}) = \mathbf{z} \oplus_c \left(\tanh\left(\sqrt{c}\frac{\lambda_c\|\mathbf{v}\|}{2}\right)\frac{\mathbf{v}}{\sqrt{c}\|\mathbf{v}\|}\right). \tag{2.15}$$

By setting the origin as the fixed base point, the exponential map can be simplified as:

$$\exp_0^c(\mathbf{v}) = \tanh\left(\sqrt{c}\|\mathbf{v}\|\right)\frac{\mathbf{v}}{\sqrt{c}\|\mathbf{v}\|}. \tag{2.16}$$

After exponential mapping, the geodesic distance between two points in the hyperbolic manifold (hyperbolic distance) can be obtained as

$$d_{\text{hyp}}(\mathbf{x}, \mathbf{y}) = \frac{2}{\sqrt{c}}\operatorname{arctanh}\left(\sqrt{c}\|-\mathbf{x} \oplus_c \mathbf{y}\|\right). \tag{2.17}$$

Hyperbolic embedding for features has been proposed for datasets that have some underlying tree structure [94]. The paper [95] derives hyperbolic versions of several deep learning tools, including multinomial logistic regression, feed-forward networks, and recurrent networks. In the field of natural language processing, [96] and [97] introduce feature embeddings with hyperbolic models. In the field of deep graph learning, HGCN [91] considers hyperbolic node embeddings in graph neural networks. GIL [98] proposes to use weighted embedding features in both Euclidean and hyperbolic spaces. In the computer vision community, [99] uses pair-wise cross-entropy loss with hyperbolic distances to train the vision transformer [25], and [92] considers hyperbolic embeddings in the semantic segmentation task. More recently, hyperbolic embeddings have also been studied for the 3D point cloud [93], where

the 3D point cloud is treated as natural compositions of small parts that follow the hierarchical architecture. This motivates us to introduce hyperbolic embeddings in our pipeline for better feature representations.

Chapter 3

Single-Modal Camera Localization

Cameras are one of the most widely used sensors for environmental perception in visual localization tasks. However, as passive sensors, they are highly sensitive to environmental changes such as lighting variations, weather conditions, and other external factors. These perturbations can often lead to significant degradation in localization accuracy, particularly in challenging and dynamic environments.

Many existing camera localization methods [2, 22, 30, 31] fail to adequately address the sensitivity of cameras to these environmental changes. As a result, their models tend to underperform in such adverse conditions. In addition, the temporal consistency between different frames is not effectively explored, which results in unsatisfactory performance in multi-frame localization scenarios.

To mitigate these challenges, we propose the integration of neural ODEs into the feature extraction pipeline. Neural ODEs naturally provide robustness that stabilizes the feature extraction process under challenging conditions. In addition to this, expanding the perception FOV is another effective strategy for improving robustness, as it incorporates additional scene information to better handle environmental perturbations. To achieve this, we employ GNNs to aggregate and integrate information from neighboring visual features. GNNs enhance the model's ability to capture spatial relationships and contextual details, which help further strengthen the model's robustness in dynamic environments. In this chapter, we present a camera localization framework that leverages both neural ODEs and GNNs. This combination results in superior pose estimation accuracy and significantly improves robustness in challenging driving scenarios.

3.1 Introduction

Camera pose regression treats visual localization as a regression problem, where the network takes an image as input and outputs the camera's 6-DoF pose relative to a given coordinate frame. This problem is critical for applications such as autonomous driving and augmented reality, where accurate and real-time localization is essential for safe navigation in dynamic environments.

The pioneering work PoseNet [2] introduces a CNN-based approach to extract features from a single image and directly regress the camera's 6-DoF pose. While this approach marks a significant milestone, it struggles in real-world scenarios, especially in environments with dynamic elements, changing lighting conditions, and seasonal variations. These environmental perturbations can degrade the performance of camera pose regression models, making them unreliable for autonomous driving in complex conditions.

To address some of these limitations, more advanced approaches have extended the input from a single image to multiple views. For example, MapNet [28] improves performance by integrating visual odometry as a post-processing step to refine the pose trajectory. However, this reliance on pre-computed odometry limits its lightweight operation. GNNMapNet [100] takes this further by leveraging GNNs to allow images to interact with their neighboring frames, but it still faces challenges in handling extreme environmental perturbations and does not fully exploit the temporal consistency available in multi-frame inputs.

To tackle these issues, we introduce neural ODEs into the feature extraction process. Neural ODEs offer intrinsic robustness, helping to stabilize feature extraction under challenging conditions. Additionally, increasing the FOV emerges as an effective strategy for enhancing robustness, as it brings in more scene information, helping to address input noise. To implement this, we leverage GNNs to aggregate information from neighboring visual features, enabling the model to capture spatial relationships and contextual details more effectively. This also expands the FOV through neighboring features, reinforcing the model's robustness in dynamic environments. By incorporating neural ODEs and GNNs, we propose RobustLoc for robust camera localization in challenging conditions.

We evaluate RobustLoc on three challenging autonomous driving datasets, including scenarios with severe weather conditions and sensor noise, and demonstrate that it significantly outperforms existing state-of-the-art (SOTA) camera pose regression models. Through extensive ablation studies, we also provide insights into the effectiveness of each component, including the contribution of neural ODEs and multi-view information aggregation.

Our main contributions are summarized as follows:

- We propose a graph-based framework for camera pose regression that integrates neural ODEs and GNNs to enhance the robustness of feature extraction under environmental perturbations. We introduce feature diffusion blocks at both the feature map and embedding stages, ensuring robust propagation of information.
- We present a multi-level training strategy with branched decoders to further refine pose regression and provide robustness across varying input conditions. Extensive experiments on noisy, real-world datasets demonstrate that RobustLoc achieves state-of-the-art performance in challenging conditions. We also conduct ablation studies to highlight the importance of each module and provide insights into the model's robustness.

3.2 Methodology

In this section, we provide a detailed description of our proposed camera pose regression approach. We assume that the input is a set of images $\{\mathbf{I}_i\}_{i\in[N^{\text{view}}]}$ that may be covisible (see Fig. 3.1).[1] Our objective is to perform camera pose regression on the input images.

3.2.1 RobustLoc Overview

We first summarize our multi-view camera pose regression pipeline, which can be decomposed into three different stages, as follows (see Fig. 3.2 and Fig. 3.3):

[1] *Notations:* We use $[N^{\text{view}}]$ to denote the set of integers $\{1, 2, \ldots, N^{\text{view}}\}$.We use boldfaced lowercase letters like $\mathbf{m}$ to denote vectors and boldface capital letters like $\mathbf{W}$ to denote matrices.

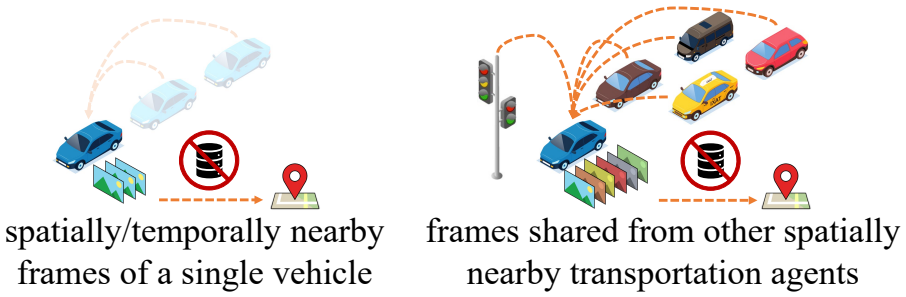

FIGURE 3.1: Multi-view camera pose regression with neighboring information, without the need for any database.

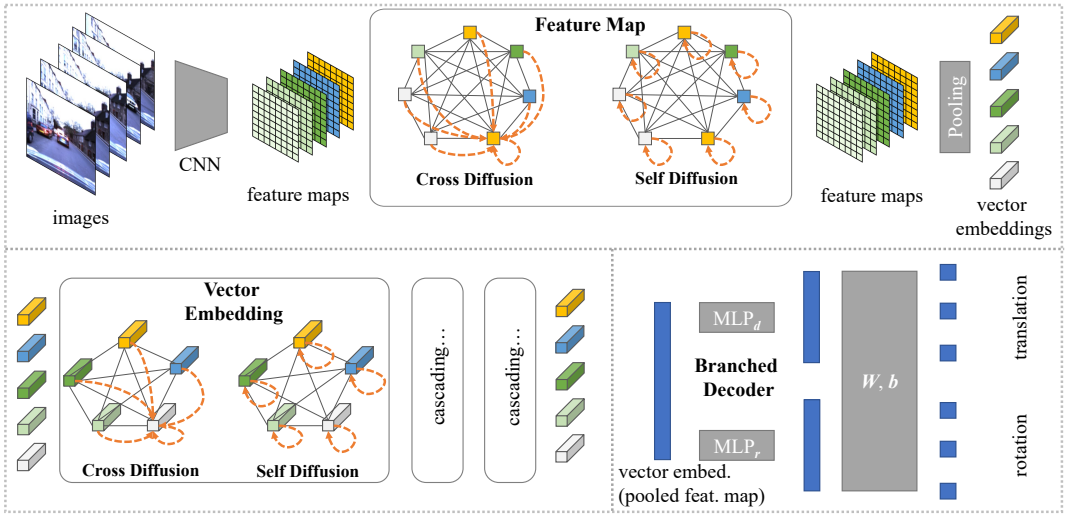

FIGURE 3.2: The main architecture of RobustLoc. Feature diffusion is performed at both the feature map stage and the vector embedding stage. The branched decoder regresses the 6-DoF poses based on the vector embeddings or the pooled feature maps. The details for multi-layer decoding are shown in Fig. 3.3.

1. Given N neighboring images, a CNN extracts the feature maps of all these images. Our proposed feature map diffusion block then performs cross-self diffusion on the feature maps.

2. After feature map diffusion, a global average pooling module aggregates the feature maps as vector embeddings, which contain global representations of these images. Similarly, those vector embeddings are then diffused by cascaded diffusion blocks.

3. Based on the vector embeddings, the branched decoder module regresses the output camera poses. During training, decoding is performed on multiple levels to provide better feature constraints.

3.2.2 Neural Diffusion for Feature Maps

The input images $\{\mathbf{I}_i\}_{i\in[N^{\text{view}}]}$ are passed through a CNN to obtain the feature maps $\{\mathbf{m}_i \in \mathbb{R}^{H\times W\times C}\}_{i\in[N^{\text{view}}]}$. Here, C is the channel dimension, while H and W are the dimensions of a feature map. For each feature map $\mathbf{m}_i$, we denote its j-th element as $\mathbf{m}_{i,j} \in \mathbb{R}^C, j \in [HW]$. We next describe the feature map diffusion block, where we perform cross-diffusion from node to node in a graph and self-diffusion within each node. The two diffusion processes update the feature map by leveraging the neighboring information or only using each node's individual information, respectively.

3.2.2.1 Cross-Diffusion Dynamics

To support the cross-diffusion over feature maps, we formulate the first graph in our pipeline as:

$$\mathcal{G}^{\text{feat}} = (\mathcal{V}^{\text{feat}}, \mathcal{E}^{\text{feat}}), \tag{3.1}$$

where the node set $\mathcal{V}^{\text{feat}} = \{\mathbf{m}_{i,j}\}_{(i,j)\in[N^{\text{view}}]\times[HW]}$ contains element-wise features $\mathbf{m}_{i,j}$ and the edge set $\mathcal{E}^{\text{feat}}$ is defined as the complete graph edges associated with attention weights as discussed below. The complete graph architecture is demonstrated to be an effective design shown in Table 3.6.

To achieve robust feature interaction, we next define the cross-diffusion process as:

$$\frac{d\mathbf{x}(t)}{dt} = f_{\text{cross}}(\mathbf{x}(t)), \tag{3.2}$$

where $f_{\text{cross}}(\mathbf{x}(t))$ is a neural network and can be approximately viewed as a neural PDE with the partial differential operations over a manifold space replaced by the attention modules that we will introduce later. We denote the input to the feature map diffusion module as the initial state at $t = t_0$ as $\mathbf{x}(t_0) = \{\mathbf{m}_{i,j}\}_{(i,j)\in[N^{\text{view}}]\times[HW]}$, where $\mathbf{x}(t) = \{\mathbf{m}_{i,j}(t)\}_{(i,j)\in[N^{\text{view}}]\times[HW]}$ denotes the hidden state of the diffusion.

The diffusion process is known to have robustness against local perturbations of the manifold, where the local perturbations in our camera pose regression task include challenging weather conditions, dynamic street objects, and unexpected image noise. Therefore, the diffusion-based module (3.2) can be simultaneously

capable of leveraging the neighboring image information and holding robustness against local perturbations.

We next introduce the computation of attention weights in $f_{\text{cross}}(\mathbf{x}(t))$ for node features at time t. We first generate the embedding of each node using multi-head fully connected (FC) layers with learnable parameter matrix $\mathbf{W}_k$ and bias $\mathbf{b}_k$ at each head $k = [K]$, where K is the number of heads. The output at each head k can be written as:

$$\mathbf{m}^{\text{FC}}_{i,j;k}(t) = \mathbf{W}_k \mathbf{m}_{i,j}(t) + \mathbf{b}_k. \tag{3.3}$$

The attention weights are then generated by computing the dot product among all the neighboring nodes using the features $\left\{\mathbf{m}^{\text{FC}}_{i,j;k}(t)\right\}_{(i,j)\in[N^{\text{view}}]\times[HW]}$. We have

$$\begin{aligned}&\left\{a_{(i,j),(i',j');k}(t)\right\}_{(i',j')\in\mathcal{N}_{i,j}}\\ &= \text{Softmax}_{(i',j')\in\mathcal{N}(i,j)}(\mathbf{m}^{\text{FC}}_{i,j;k}(t)\cdot\mathbf{m}^{\text{FC}}_{i',j';k}(t)),\end{aligned} \tag{3.4}$$

where $\mathcal{N}_{i,j}$ denotes the set of neighbors of node $\mathbf{m}_{i,j}$. Let

$$\mathbf{m}^{\text{weighted}}_{i,j;k}(t) = \sum_{(i',j')\in\mathcal{N}_{i,j}} a_{(i,j),(i',j');k}(t)\mathbf{m}^{\text{FC}}_{i',j';k}(t). \tag{3.5}$$

Finally, the updated node features are obtained by concatenating the weighted node features from all heads as

$$f_{\text{cross}}(\mathbf{x}(t)) = \left\{ \Big\|_{k\in[K]} (\mathbf{m}^{\text{weighted}}_{i,j;k}(t)) \right\}_{(i,j)\in[N^{\text{view}}]\times[HW]}. \tag{3.6}$$

Based on the above pipeline, the output of the cross-diffusion at time $t = t_1$ can be obtained as:

$$\mathbf{x}(t_1) = F_{\text{cross}}(\mathbf{x}(t_0)), \tag{3.7}$$

where $F_{\text{cross}}(\,\cdot\,)$ denotes the solution of (3.2) integrated from $t = t_0$ to $t = t_1$.

3.2.2.2 Self-Diffusion Dynamics

In the next step, we update each node feature independently. The node-wise feature update can be regarded as a rewiring of the complete graph to an edgeless graph, and the node-wise feature update is described as:

$$\frac{d\mathbf{m}_{i,j}(t)}{dt} = f_{\text{self}}(\mathbf{m}_{i,j}(t)) = \text{MLP}(\mathbf{m}_{i,j}(t)). \tag{3.8}$$

And the output of self-diffusion can be obtained as:

$$\mathbf{m}_{i,j}(t_2) = F_{\text{self}}(\mathbf{m}_{i,j}(t_1)). \tag{3.9}$$

where $F_{\text{self}}(\cdot)$ denotes the solution of (3.8) integrated from $t = t_1$ to $t = t_2$. As neural ODEs are robust against input perturbations [78], the updating of each node feature according to the self-diffusion (3.8) can be robust against perturbations like challenging weather conditions, dynamic street objects, and image noise.

3.2.3 Vector Embeddings and Diffusion

After the feature map neural diffusion, we feed the updated feature maps into a global average pooling module to generate the vector embeddings $\{\mathbf{h}_i \in \mathbb{R}^C\}_{i\in[N^{\text{view}}]}$, where

$$\mathbf{h}_i = \text{Pooling}(\mathbf{m}_i). \tag{3.10}$$

Each vector embedding contains rich global representations for the input image together with the information diffused from the neighboring images. To enable diffusion for the global information, we propose to design the vector embedding graph as:

$$\mathcal{G}^{\text{vect}} = (\mathcal{V}^{\text{vect}}, \mathcal{E}^{\text{vect}}), \tag{3.11}$$

where the node set $\mathcal{V}^{\text{vect}} = \{\mathbf{h}_i\}_{i\in[N^{\text{view}}]}$ contains image vector embeddings $\mathbf{h}_i$ and the edge set $\mathcal{E}^{\text{vect}}$ is also defined to be the complete graph. Based on this graph $\mathcal{G}^{\text{vect}}$, we construct the cascaded diffusion blocks, to perform global information diffusion. Within the cascaded blocks, each basic diffusion block consists of two diffusion layers: a cross-diffusion layer and a self-diffusion layer, similar to the two diffusion schemes introduced at the feature map diffusion phase.

3.2.4 Pose Decoding

In this subsection, we explain the pose decoding operations.

3.2.4.1 Branched Pose Decoder

Each camera pose $\{\mathbf{d}, \mathbf{r}\} \in \mathbb{R}^6$, consists of a 3-dimensional translation $\mathbf{d} \in \mathbb{R}^3$ and a 3-dimensional rotation $\mathbf{r} \in \mathbb{R}^3$. Thus camera pose regression can be viewed as a multi-task learning problem. However, since the translation and rotation elements do not scale compatibly, the regression converges in different basins. To deal with it, previous methods consider regression for translation and rotation respectively. In our paper, we also follow this insight to design the decoder.

Firstly, the feature embeddings $\{\mathbf{h_d}, \mathbf{h_r}\}$ for translation and rotation are extracted from the feature embedding $\mathbf{h}$ using different non-linear MLP layers as:

$$\mathbf{h_d} = \text{MLP}_{\mathbf{d}}(\mathbf{h}), \tag{3.12}$$

$$\mathbf{h_r} = \text{MLP}_{\mathbf{r}}(\mathbf{h}), \tag{3.13}$$

Thus, the features of translation and rotation are decoupled. Next in the second stage, the pose output can be regressed as:

$$\text{pose} = \mathbf{W}(\mathbf{h_d} \,\big\|\, \mathbf{h_r}) + \mathbf{b} \tag{3.14}$$

where $\mathbf{W}, \mathbf{b}$ are learnable parameters. During training, we compute the regression loss of decoded poses from multiple levels, which we will introduce below. During inference, we use the decoded pose from the last layer as the final output pose.

3.2.4.2 Multi-level Pose Decoding Graph

To better regularize the whole regression pipeline, we propose to leverage the feature maps at multiple levels. As shown in Fig. 3.3, at the vector embedding stage, we use the vector embeddings to regress the poses, while at the feature map stage, we use the feature maps. Denoting the feature maps at layer l as $\{\mathbf{m}_i^l \in \mathbb{R}^{H\times W\times C}\}_{i\in[N^{\text{view}}]}$, the pose decoding graph at layer l can be formulated as:

$$\mathcal{G}^{\text{pose},l} = (\mathcal{V}^{\text{pose},l}, \mathcal{E}^{\text{pose},l}), \tag{3.15}$$

where edge set $\mathcal{E}^{\text{pose},l}$ is defined to be connected with two spatially adjacent nodes which can be viewed as the odometry connection, while the node set $\mathcal{V}^{\text{pose},l}$ is defined depending on layers since the information used to regress poses is different:

$$\mathcal{V}^{\text{pose},l} = \begin{cases} \{\mathbf{h}_i\}_{i\in[N^{\text{view}}]} & \text{if } l = L, \\ \{\mathbf{m}_i^l\}_{i\in[N^{\text{view}}]} & \text{otherwise,} \end{cases} \tag{3.16}$$

where L represents the last layer in our network.

At the last layer where there are vector embeddings, we can directly apply the pose decoder to generate absolute pose messages. By contrast, at feature map layers, we first apply a global average pooling module on the feature maps to formulate feature vectors, and pose messages can be obtained using the pose decoder:

$$\text{pose}_i^l = \begin{cases} f_{\text{decoder}}^l(\mathbf{h}_i^l) & \text{if } l = L, \\ f_{\text{decoder}}^l(\text{Pooling}(\mathbf{m}_i^l)) & \text{otherwise.} \end{cases} \tag{3.17}$$

where $f_{\text{decoder}}^l(\cdot)$ is the pose decoder at layer l. Using the simplified relative pose computation technique, the relative pose messages $\text{pose}_{i,i'}^l$ at layer l can be generated as:

$$\text{pose}_{i,i'}^l = \text{pose}_{i'}^l - \text{pose}_i^l. \tag{3.18}$$

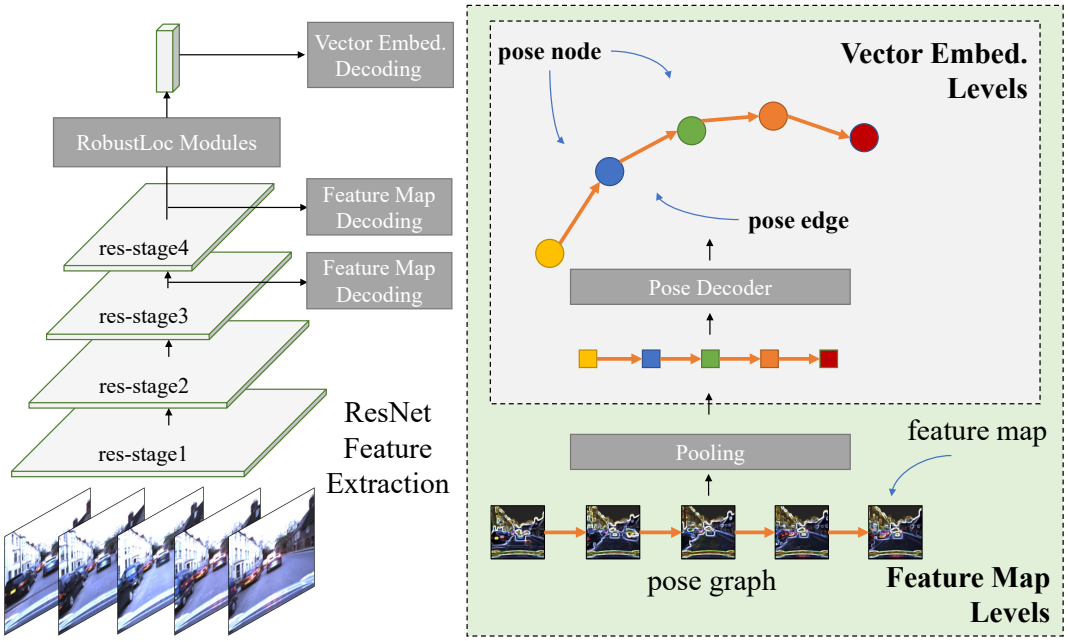

FIGURE 3.3: Multi-level pose decoding. Decoding can be directly applied to vector embeddings. Feature maps are first pooled and then decoded.

By leveraging multi-layer information, not only the last layer but also the preceding middle-level layers can directly learn the implicit relation between images and poses, which helps to improve the robustness against perturbations.

3.2.5 Loss Function

We use a weighted balance loss for translation and rotation predictions. For the i-th input image, we denote the translation and rotation targets as $\mathbf{d}_i^* \in \mathbb{R}^3$ and $\mathbf{r}_i^* \in \mathbb{R}^3$ respectively. Then the absolute pose loss term $\mathcal{L}_i^l$ and the relative pose loss term $\mathcal{L}_{i,i'}^l$ at decoding layer l are computed as:

$$\begin{aligned}\mathcal{L}_i^l &= \left\|\mathbf{d}_i^l - \mathbf{d}_i^*\right\| \exp(-\alpha^l) + \alpha^l \\ &\quad + \left\|\mathbf{r}_i^l - \mathbf{r}_i^*\right\| \exp(-\beta^l) + \beta^l,\end{aligned} \tag{3.19}$$

$$\begin{aligned}\mathcal{L}_{i,i'}^l &= \left\|\mathbf{d}_{i,i'}^l - \mathbf{d}_{i,i'}^*\right\| \exp(-\rho^l) + \rho^l \quad , \\ &\quad + \left\|\mathbf{r}_{i,i'}^l - \mathbf{r}_{i,i'}^*\right\| \exp(-\lambda^l) + \lambda^l\end{aligned} \tag{3.20}$$

where $\mathbf{d}_i^l, \mathbf{r}_i^l, \mathbf{d}_{i,i'}^l, \mathbf{r}_{i,i'}^l$ are outputs at layer l, while $\alpha^l, \beta^l, \rho^l, \lambda^l$ are all learnable parameters at layer l. Finally, the overall loss function can be obtained as:

$$\mathcal{L} = \sum_{l \in \{3,4,L\}} \sum_{i \in [N^{\text{view}}], i' \in \mathcal{N}_i^l} \mathcal{L}_i^l + \mathcal{L}_{i,i'}^l, \tag{3.21}$$

where $\mathcal{N}_i^l$ is the neighborhood of node i in $\mathcal{G}^{\text{pose},l}$.

3.3 Experiments

In this section, we first evaluate our proposed model on three large autonomous driving datasets. We next present an ablation study to demonstrate the effectiveness of our model design.

3.3.1 Datasets and Implementation Details

3.3.1.1 Oxford RobotCar

The Oxford RobotCar dataset[101] is a large autonomous driving dataset collected by a car driving along a route in Oxford, UK. It consists of two different routes: 1) Loop with a trajectory area of $8.8 \times 10^4\text{m}^2$ and length of 10^3m, and 2) Full with a trajectory area of $1.2 \times 10^6\text{m}^2$ and length of 9×10^3m.

3.3.1.2 4Seasons

There are only a few existing methods designed for robust camera pose regression in driving environments, and the experiment on the Oxford dataset is insufficient for comparison. Thus we also conduct experiments on another driving dataset to cover more driving scenarios. The 4Seasons dataset [102] is a comprehensive dataset for autonomous driving. It was collected in Munich, Germany, covering varying perceptual conditions. Specifically, it contains different environments including the business area, the residential area, and the town area. In addition, it consists of a wide variety of weather conditions and illuminations. In our experiments, we use 1)

Business Campus (business area), 2) Neighborhood (residential area), and 3) Old Town (town area).

3.3.1.3 Perturbed RobotCar

To further evaluate the performance under challenging environments, we inject noise into the RobotCar Loop dataset and call this the Perturbed RobotCar dataset as shown in Fig. 3.4. We create three scenarios: 1) Medium (with a mixture of noises: fog, snow, rain, and spatter on the lens), 2) Hard (with added Gaussian noise), and 3) Hard (+ *noisy training*) (i.e., training with noisy augmentation). All augmentations are implemented using the imgaug package [2].

FIGURE 3.4: Visualization of the Perturbed RobotCar dataset. Medium is with a mixture of noises: fog, snow, rain, and spatter on the lens. Hard is with added Gaussian noise.

3.3.1.4 Implemention Details

We use ResNet34 as the backbone, which is pre-trained on the ImageNet dataset. We set the maximum number of input images as 11. We resize the shorter side of each input image to 128 and set the batch size to 64. The Adam optimizer with a learning rate 2×10^{-4} and weight decay 5×10^{-4} is used to train the network.

[2] `https://github.com/aleju/imgaug`

	Model	Loop (cross-day) Mean	Loop (cross-day) Median	Loop (within-day) Mean	Loop (within-day) Median	Full Mean	Full Median
+ Extra	GNNMapNet + *post.*[100]	7.96 / 2.56	-	-	-	17.35 / 3.47	-
	ADPoseNet[29]	-	-	-	6.40 / 3.09	-	33.82 / 6.77
	ADMapNet[29]	-	-	-	6.45 / 2.98	-	19.18 / 4.60
	MapNet+[28]	8.17 / 2.62	-	-	-	30.3 / 7.8	
	MapNet+ + *post.*[28]	6.73 / **2.23**	-	-	-	29.5 / 7.8	-
Pure Models	GeoPoseNet[22]	27.05 / 18.54	6.34 / 2.06	-	-	125.6 / 27.1	107.6 / 22.5
	MapNet[28]	9.30 / 3.71	5.35 / 1.61	-	-	41.4 / 12.5	17.94 / 6.68
	LsG[103]	9.08 / 3.43	-	-	-	31.65 / 4.51	-
	AtLoc[30]	8.74 / 4.63	5.37 / 2.12	-	-	29.6 / 12.4	11.1 / 5.28
	AtLoc+[30]	7.53 / 3.61	4.06 / 1.98	-	-	21.0 / 6.15	6.40 / 1.50
	CoordiNet[31]	-	-	4.06 / 1.44	2.42 / 0.88	14.96 / 5.74	**3.55** / 1.14
	RobustLoc (ours)	**4.68** / 2.67	**3.70** / **1.50**	**2.49** / **1.40**	**1.97**/ **0.84**	**9.37** / **2.47**	5.93 / **1.06**

TABLE 3.1: Median and mean translation/rotation estimation error (m/°) on the Oxford RobotCar dataset. The best and the second-best results in each metric are highlighted in **bold** and underlined respectively. "-" denotes no data provided.

Data augmentation techniques include random cropping and color jittering. We set the integration times $t_0 = 0$, $t_1 = 1$, and $t_2 = 2$. The number of attention heads is 8. We train our network for 300 epochs. All of the experiments are conducted on an NVIDIA A5000.

3.3.2 Main Results

On the Oxford RobotCar dataset, as shown in Table 3.1, we obtain the best performance in 10 out of 12 metrics. Using the mean error, which is easily influenced by outlier predictions, RobustLoc outperforms the baselines by a significant margin. In the most challenging route Full, to the best of our knowledge, RobustLoc is the first to achieve less than 10m mean translation error for camera pose regression.

The 4Seasons dataset consists of more varied driving scenes. As shown in Table 3.2, RobustLoc achieves the best performance in 11 out of 12 metrics. Again, using the mean error metric, RobustLoc outperforms the baselines by a significant margin.

On the Perturbed RobotCar dataset, where the images contain more challenging weather conditions and noisy perturbations, RobustLoc achieves the best in all metrics. The superiority of RobustLoc over other baselines is more obvious in Table 3.3.

Model	Business Campus		Neighborhood		Old Town	
	Mean	Median	Mean	Median	Mean	Median
GeoPoseNet[22]	11.04 / 5.78	5.93 / 2.03	2.87 / 1.30	1.92 / 0.88	64.81 / 6.67	15.03 / 1.57
MapNet[28]	10.35 / 3.78	5.66 / 1.83	2.81 / 1.05	1.89 / 0.92	46.56 / 7.14	16.52 / 2.12
GNNMapNet[100]	7.69 / 4.34	5.52 / 2.16	3.02 / 2.92	2.14 / 1.45	41.54 / 7.30	19.23 / 3.26
AtLoc[30]	11.53 / 4.84	5.81 / 1.50	2.80 / 1.16	1.83 / 0.93	84.17 / 7.81	17.10 / 1.73
AtLoc+[30]	13.70 / 6.41	5.58 / 1.94	2.33 / 1.39	1.61 / 0.88	68.40 / 5.51	14.52 / 1.69
IRPNet[104]	10.95 / 5.38	5.91 / 1.82	3.17 / 2.85	1.98 / 0.90	55.86 / 6.97	17.33 / 3.11
CoordiNet[31]	11.52 / 3.44	6.44 / **1.38**	1.72 / 0.86	1.37 / 0.69	43.68 / 3.58	11.83 / 1.36
RobustLoc (ours)	**4.28** / **2.04**	**2.55** / 1.50	**1.36** / **0.83**	**1.00** / **0.65**	**21.65** / **2.41**	**5.52** / **1.05**

TABLE 3.2: Median and mean translation/rotation estimation error (m/°) on the 4Seasons dataset. The best and the second-best results in each metric are highlighted in **bold** and underlined respectively.

Model	Medium		Hard		Hard (+ *noisy training*)	
	Mean	Median	Mean	Median	Mean	Median
GeoPoseNet[22]	20.47 / 8.76	8.70 / 2.30	41.71 / 17.63	14.02 / 3.13	24.03 / 11.14	7.14 / 1.70
MapNet[28]	17.93 / 7.01	6.89 / 2.00	49.36 / 20.01	18.37 / 2.58	21.22 / 8.38	6.38 / 1.97
GNNMapNet[100]	16.17 / 7.24	8.02 / 2.35	73.97 / 35.57	61.47 / 19.73	14.55 / 7.62	6.69 / 1.57
AtLoc[30]	19.92 / 7.25	7.26 / 1.74	52.56 / 23.46	15.01 / 3.17	23.48 / 11.43	7.42 / 2.38
AtLoc+[30]	17.68 / 7.48	6.19 / 1.80	37.92 / 18.65	12.17 / 2.93	22.61 / 11.23	6.21 / 1.83
IRPNet[104]	16.35 / 7.56	8.71 / 2.28	45.72 / 21.84	17.99 / 3.50	24.73 / 11.20	6.73 / 1.82
CoordiNet[31]	17.67 / 6.66	7.63 / 1.79	44.11 / 16.42	17.21 / 2.70	24.06 / 12.27	6.25 / 1.61
RobustLoc (ours)	**8.12** / **3.83**	**5.34** / **1.53**	**27.75** / **9.70**	**11.59** / **2.64**	**10.06** / **4.95**	**5.18** / **1.43**

TABLE 3.3: Median and mean translation/rotation estimation error (m/°) on the Perturbed RobotCar dataset. The best and the second-best results in each metric are highlighted with **bold** and underline respectively. RobustLoc achieves the best in **all** metrics.

3.3.3 Analysis

3.3.3.1 Ablation Studies

We justify our design for RobustLoc by ablating each module. From Table 3.4, we observe that every module in our design contributes to the final improved estimation. We see that making use of neighboring information from covisible frames and learning robust feature maps contribute to more accurate camera pose regression.

3.3.3.2 Diffusion Modules

We also test different combinations of diffusion modules in Table 3.5. The order of cross-diffusion and self-diffusion does not impact the performance significantly. Among them, the cross-diffusion part would contribute more than the self-diffusion

Method	Mean Error (m/°) on Loop (c.)
base model	8.38 / 4.29
+ feature map graph	7.01 / 3.86
+ vector embedding graph	6.24 / 3.21
+ diffusion	5.53 / 2.95
+ branched decoder	5.14 / 2.79
+ multi-level decoding	**4.68** / **2.67**
diffusion at stage 3	5.27 / 2.90
diffusion at stage 3,4	4.86 / 3.18
diffusion at stage 4	**4.68** / **2.67**
multi-layer concatenation	5.80 / 3.26
more augmentation	**4.68** / **2.67**
less augmentation	5.32 / 3.17

TABLE 3.4: Ablation study, diffusion design, and augmentation design comparison on the Oxford RobotCar dataset.

part, as it leverages neighboring frames that consist of more information than a single frame.

Method	Mean Error (m/°) on Loop (c.)
self-diffusion only	6.18 / 3.27
cross-diffusion only	5.47 / 3.02
self-diffusion. + cross-diffusion	4.77 / 2.69
cross-diffusion + self-diffusion	**4.68** / **2.67**

TABLE 3.5: Results of using different diffusion modules.

3.3.3.3 Salience Visualization

Salience maps shown in Fig. 3.5 suggest that in driving environments, RobustLoc pays more attention to relatively robust features such as the skyline and the road. In addition, dynamic objects such as vehicles are implicitly suppressed in RobustLoc's regression pipeline.

3.3.3.4 Diffusion and Augmentation

Using multi-level features is an effective method in dense prediction tasks such as depth estimation. To test if this holds in camera pose regression, we use the

FIGURE 3.5: Robust features from RobustLoc.

feature maps from the lower stage 3 (see Fig. 3.3), which however does not lead to performance improvement shown in Table 3.4. We also utilize the multi-level concatenation strategy used in GNNMapNet. This does not lead to significant changes. These experiments demonstrate that camera pose regression benefits more from high-level features with more semantic information than from low-level local texture features. Finally, we test the performance when training with less data augmentation, which leads to worse performance. This suggests that more extensive data augmentation can enhance the model robustness in challenging scenarios, which is consistent with the experimental results on the Perturbed Robotcar dataset in Table 3.3.

3.3.3.5 Graph Design

We next explore the use of different graph designs for feature map diffusion and vector embedding diffusion. The grid graph stacks an image with two other spatially adjacent images as a cube, and the attention weights are formulated within the 6-neighbor area (for feature maps) or the 2-neighbor area (for vector embeddings). The self-cross graph computes attention weights first within each image and then across different images. From Table 3.6, we see that the complete graph has the best performance. This is because, in the complete graph, each node can interact with all other nodes, allowing the aggregation of useful information with appropriate attention weights.

3.3.3.6 Rotation Representation

We compare different representations of rotation in Table 3.6, where the log form of the quaternion is the optimal choice. The other three representations, including

Method	Mean Error (m/°) on Full
grid graph	15.67 / 2.95
self-cross graph	15.31 / 3.28
complete graph	**9.37** / **2.47**
	Mean Error (°) on Business Campus
quaternion	2.23
Lie group	2.20
rotation matrix	2.25
log (quaternion)	**2.04**

TABLE 3.6: Graph design comparison on the Oxford RobotCar dataset and rotation representation comparison on the 4Seasons dataset.

the vanilla quaternion, the Lie group, and the vanilla rotation matrix, show similar performance.

3.3.3.7 Trajectory Visualization

We visualize the output pose trajectories as shown in Fig. 3.6, where a significant gap can be seen from the comparison. RobostLoc outputs more smooth and globally accurate poses compared with the previous method, which shows the effectiveness of our design.

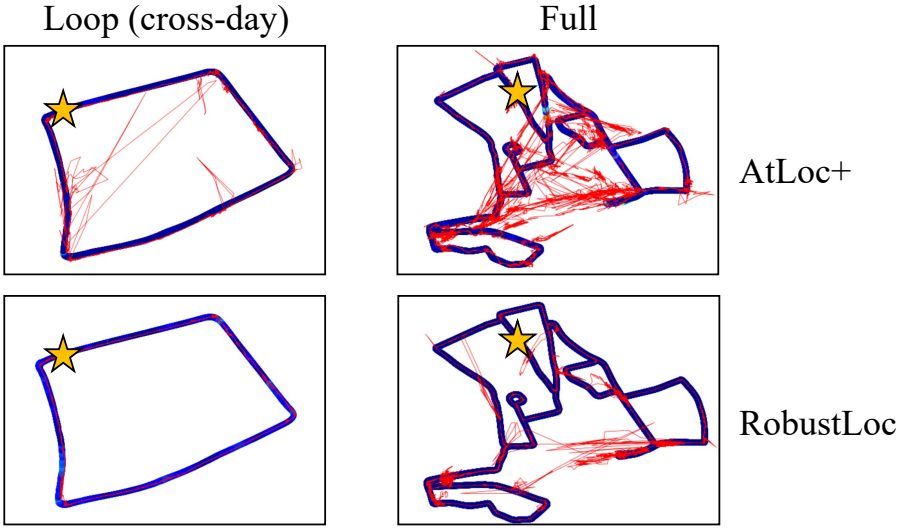

FIGURE 3.6: Trajectory visualization on the Oxford RobotCar dataset. The ground truth trajectories are shown in bold blue lines, and the estimated trajectories are shown in thin red lines. The stars mark the start of the trajectories.

#frames	3	5	7	9	11
Speed (iters/s)	**56**	55	53	52	50
Mean Error (m)	5.28	5.09	4.96	**4.68**	4.72

TABLE 3.7: The performance using different numbers of frames on the Oxford RobotCar Loop (cross-day).

3.3.3.8 Inference Speed

We finally test the performance using a different number of input frames. The inference speed does not drop significantly when increasing the input frames. And even the slowest one (using 11 frames) can run 50 iterations per second and achieve real-time regression. On the other hand, more frames can bring performance improvement when the input size is small, while further increasing frame size does not bring significant change.

3.4 Conclusion

We have proposed and verified the performance of a robust camera pose regression model RobustLoc. The model's robustness derives from the use of information from covisible images and neural graph diffusion to aggregate neighboring information, which is present in challenging driving environments. Extensive experimental results demonstrate that RobustLoc achieves SOTA performance.

Chapter 4

Single-Modal LiDAR Localization

In Chapter 3, we introduce RobustLoc, a pose regression network that takes camera images as input. While cameras are widely used for their rich semantic information and high spatial resolution, they are passive sensors that rely heavily on illumination. As a result, camera-based localization systems are highly susceptible to illumination changes, which can lead to significant performance degradation under varying lighting conditions.

By contrast, LiDARs serve as another basic key sensor for visual localization. Unlike cameras, which passively capture images, LiDARs actively emit laser scans to gather spatial data from the environment, making them less susceptible to environmental changes such as variations in illumination. However, the produced 3D point clouds present unique challenges due to their unordered and sparse nature, which is fundamentally different from the structured 2D images commonly used in localization tasks.

Effectively utilizing point cloud data for localization remains a challenge. Existing approaches [3, 38] primarily focus on point-wise networks to extract features from point clouds but often overlook the inherent 2D projection properties of these data. Furthermore, these approaches typically restrict feature extraction to Euclidean space, leaving the potential insights from non-Euclidean spaces under-explored. Addressing these gaps is crucial for improving the performance of LiDAR-based visual localization.

To address these challenges, we employ two distinct feature extraction backbones for point clouds: a vanilla point-wise backbone and a spherical projection-based

backbone. This dual approach enables the exploration of both explicit 3D spatial information and implicit 2D projection properties. Additionally, we embed the extracted features into both Euclidean and hyperbolic spaces, allowing us to capture richer spatial structures in point clouds. This multi-space feature embedding further enhances the model's ability to fully leverage the spatial information for effective visual localization.

4.1 Introduction

In Chapter 3, we present RobustLoc, a camera-based pose regression network. However, while cameras provide rich RGB visual information, they are highly sensitive to environmental factors such as low illumination, light reflections, and adverse weather conditions, which can degrade the accuracy of localization. Cameras also capture dense and textured data that can be challenging to process when the environmental conditions are unfavorable.

In contrast, LiDAR sensors provide a more robust alternative, as they actively emit laser beams to measure the depth and distance of surrounding objects, making them less vulnerable to changes in environmental conditions. LiDARs are particularly effective in low-light environments or situations where visual details are sparse. Despite these advantages, LiDAR data comes in the form of sparse, unordered point clouds, which pose significant challenges for feature extraction in localization tasks.

Existing LiDAR-based pose regression networks, such as PointLoc [3] and Memory-Aware LiDAR Localization [38], primarily rely on point-based feature extraction methods, which treat each point in the cloud as an independent entity. While this approach works well in certain situations, it fails to account for the spatial structure and relationships between neighboring points, particularly when considering the neighborhood in a 2D projection format. This can lead to suboptimal performance in complex environments where geometric information is crucial for localization. Furthermore, most existing methods only operate within Euclidean space, limiting their ability to explore the intricate geometric properties of point clouds that may reside in non-Euclidean spaces.

To address these limitations, we propose HypLiLoc, a LiDAR-based pose regression network that incorporates a more comprehensive approach to feature extraction.

HypLiLoc is designed to handle the sparsity and unstructured nature of point clouds while capturing both local and global geometric features from multiple perspectives. Our method features a parallel feature extraction design, where point cloud features are learned in both 3D and 2D spaces. Specifically, we apply a 2D spherical projection of the point cloud, which allows us to capture neighborhood relationships that are often overlooked by purely point-based methods.

Moreover, to fully leverage the geometric properties of the point cloud, we embed the features into both Euclidean and hyperbolic spaces. This multi-space learning framework allows us to exploit the curvature of the underlying feature manifold, thereby enabling HypLiLoc to capture richer geometric information from the point cloud. The inclusion of hyperbolic space is particularly important for representing hierarchical and structured data, as it provides a more natural space for encoding features with complex, non-linear relationships.

Our main contributions are summarized as follows:

- A novel LiDAR-based pose regression network, HypLiLoc: HypLiLoc features a dual-backbone design, with one branch extracting 3D features directly from the point cloud and another extracting 2D features from a spherical projection. This parallel design enables the network to capture both local and global spatial relationships in the data.
- Multi-space learning for feature extraction: We introduce a novel multi-space learning approach that embeds features into both Euclidean and hyperbolic spaces. By leveraging both types of spaces, we capture a wider range of geometric properties from the point cloud, enabling more accurate and robust pose estimation.

4.2 Methodology

In this section, we provide a detailed description of our proposed approach. We first summarize the HypLiLoc pipeline as follows.

1. Given a LiDAR point cloud scan, in addition to the traditional backbone of extracting 3D features from the point cloud, we additionally project the 3D

points into a sphere to generate a 2D projection image. These two types of features are extracted by separate backbones.

2. We merge the two modal features together as the fusion features. The fusion features are then embedded in both Euclidean and hyperbolic spaces to achieve more effective representations.

3. After features interact in different spaces and modalities, the global feature vector is obtained by applying the global average pooling operation on the fusion features. The final pose prediction is generated using the global feature vector with the pose regression head.

4.2.1 Modal-Specific Backbones

Projection Feature Extraction. Multi-modal feature extraction has shown promising performance in various tasks [105]. The point cloud generated by LiDARs is convertible into multiple modalities by projecting 3D points into specific 2D spaces. Each projection provides us with a different way to define the neighbors of a point so that the point can aggregate feature representations from different definitions of its "neighborhood" (Fig. 4.1). To this end, we consider two typical projection methods, including the spherical projection and the BEV projection.

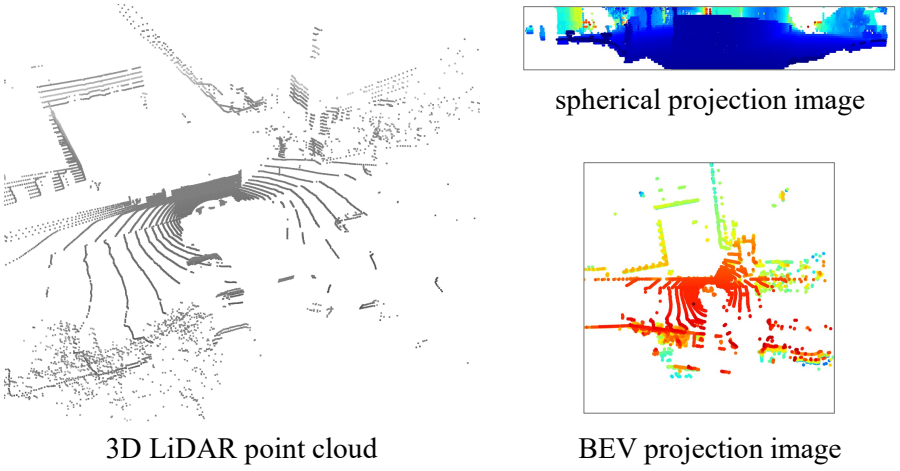

FIGURE 4.1: Visualization of the spherical and BEV projection methods.

We test the two projection counterparts in Section 4.3.4, where the spherical projection performs better than the BEV. The reason could be that LiDARs

operate with the spinning mechanism, which is better modeled by the spherical projection. By contrast, the BEV projection loses information as some points are stacked on the same pixel and is thus not a bijective mapping.

In the following discussion, we use the spherical projection strategy in our pipeline. We treat the spherical projection points as the image modality input, while the 3D features are extracted directly from the point cloud in a separate backbone that we will introduce later. Following the golden rule for 2D image processing, we use ResNet [23] as the backbone for the image modality. Denoting the ResNet backbone as $f^{\text{sph}}(\cdot)$, the final spherical projection features $\mathbf{F}^{\text{sph}} \in \mathbb{R}^{H^{\text{sph}} \times W^{\text{sph}} \times C}$ are obtained as:

$$\mathbf{F}^{\text{sph}} = f^{\text{sph}}\left(\mathbf{I}^{\text{sph}}\right). \tag{4.1}$$

3D Feature Extraction. Effective 3D point feature extraction is critical in the model design. PointNet++ [37] has shown promising performance in various tasks [3]. In our pipeline, we use PointNet++ as the backbone branch for 3D feature extraction.

PointNet++ only considers the neighboring information within a determined range, i.e., in the set abstraction (SA) layer, each centroid uses the maximum neighboring feature value as its updated feature as shown in Fig. 4.2.

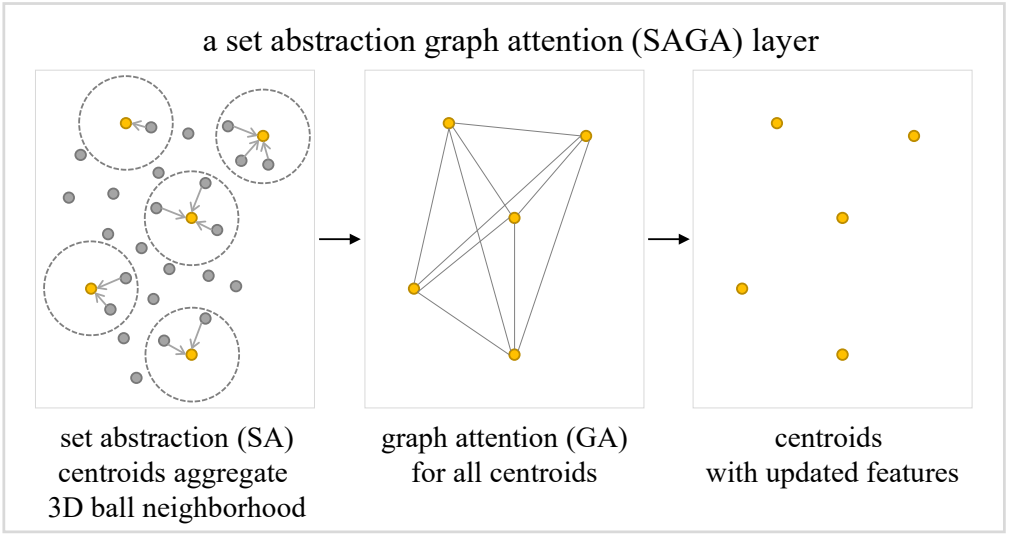

FIGURE 4.2: The SAGA layer consists of a SA layer and a GA layer.

In pose regression, the estimation accuracy of the pose benefits from an effective global representation. Thus to enable PointNet++ to additionally aggregate more global information, we introduce an additional graph attention (GA) layer after each SA layer to build the set abstraction graph attention (SAGA) layer as shown in Fig. 4.2. In this GA layer, we construct a complete graph whose node set contains all centroids from the SA layer. We denote the output of the SA layer as $\mathbf{F}^{\mathrm{SA}} \in \mathbb{R}^{N^{\mathrm{SA}} \times C^{\mathrm{SA}}}$ with N^{SA} centroids, each with a C^{SA}-dimensional feature vector. We first use a Fully-Connected (FC) layer to generate the multi-head features:

$$\mathbf{F}_k^{\mathrm{SAFC}} = \mathbf{F}^{\mathrm{SA}} \mathbf{W}_k + \mathbf{b}_k, \tag{4.2}$$

where $\mathbf{W}_k$ and $\mathbf{b}_k$ are a linear operation and an additive bias, respectively. These are learnable parameters of the k-th head. Then the attention weight matrix $\mathbf{A}_k \in \mathbb{R}^{N^{\mathrm{SA}} \times N^{\mathrm{SA}}}$ can be obtained by computing the dot product among all the neighboring nodes:

$$\mathbf{A}_k = \mathrm{Softmax}\left(\mathbf{F}_k^{\mathrm{SAFC}} \mathbf{F}_k^{\mathrm{SAFC}\intercal}\right), \tag{4.3}$$

where $\mathrm{Softmax}(\cdot)$ denotes the row-wise softmax function. The output features of the GA layer $\mathbf{F}^{\mathrm{GA}} \in \mathbb{R}^{N^{\mathrm{GA}} \times C^{\mathrm{GA}}}$ are generated by concatenating the weighted features from all heads as:

$$\mathbf{F}^{\mathrm{GA}} = \Big\Vert_k \mathbf{A}_k \mathbf{F}_k^{\mathrm{SAFC}}, \tag{4.4}$$

where $\|$ denotes the concatenation operation. We then stack L^{3D} such SAGA layers to build the 3D features extraction backbone. We denote the final output features as $\mathbf{F}^{\mathrm{3D}} \in \mathbb{R}^{N^{\mathrm{3D}} \times C}$ shown in Fig. 4.3.

4.2.2 Hyperbolic Feature Learning

The two-branch design introduced in Section 4.2.1 can be regarded as a combination of both Euclidean (vanilla 3D backbone) and spherical (2D spherical projection backbone) spaces with zero and positive curvatures respectively. We can further expand to explore hyperbolic space with negative curvatures to further exploit various features. In this subsection, we first state the motivation for such hyperbolic feature learning. We then introduce the hyperbolic embedding operators that will

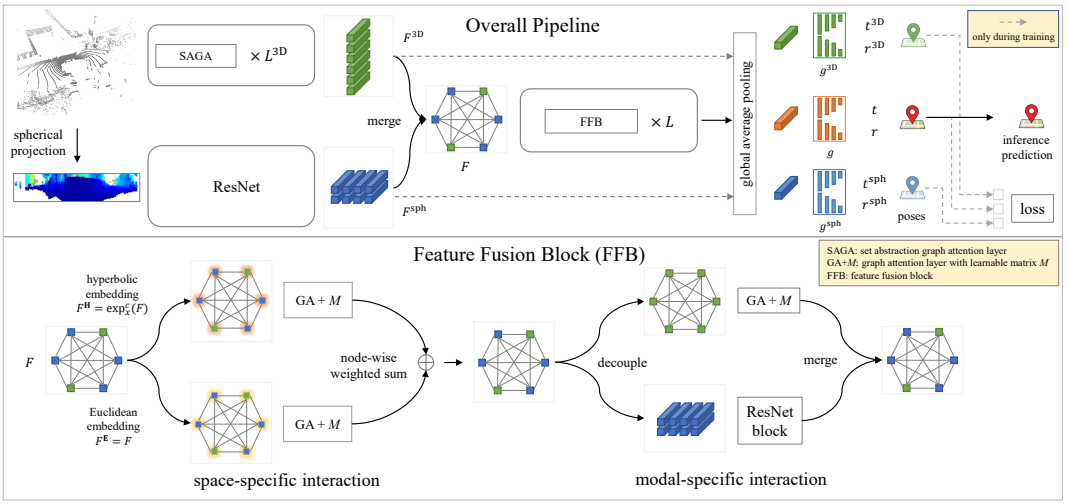

FIGURE 4.3: The overall architecture of our proposed HypLiLoc. We use two backbone branches to perform feature extraction. In the 3D backbone, we consider both local set abstraction and global attention aggregation. In the feature fusion block, the extracted multi-modal features are embedded into both Euclidean and hyperbolic spaces to achieve space-specific interaction. The fusion features are then decoupled to their own modality to perform modal-specific interaction. The final training loss is applied on both the 3D/projection level and the final fusion level.

be used in Section 4.2.3 to fuse features extracted from the 3D LiDAR point cloud and spherical LiDAR projection in Section 4.2.1.

Motivation. After feature extraction using the two backbone branches, we need an effective fusion strategy to consider both point features and projection features. Embedding in a hyperbolic space has recently gained increased interest and shown promising performance in various fields [91, 92]. The paper [93] argues that 3D point cloud objects possess inherent hierarchies due to their nature as compositions of small parts, which can be embedded in the hyperbolic space. Following this motivation, we consider leveraging the hyperbolic embedding method in our pipeline, such that features can be equipped with more various representations that come from different embedding spaces. Our ablation study (cf. Table 4.3 of Section 4.3.4) also indicates that hyperbolic embedding can lead to improvements in the pose estimation accuracy.

4.2.3 Feature Fusion Block

Based on the above-mentioned motivation, we propose the feature fusion block (FFB) to achieve effective feature interaction. Each FFB conducts both space-specific interaction and modal-specific interaction alternatively, which is similar to the commonly used cross-self-attention operation. We stack L such FFBs in our pipeline.

Feature Merging. Given the extracted 3D features $\mathbf{F}^{\text{3D}}$ and spherical projection features $\mathbf{F}^{\text{sph}}$, we first pass them through an ℓ_2 normalization layer such that all features are constrained on a sphere. This is a common way to process multi-modal data [105]. We then formulate a fusion graph with complete edge connections, in which each node contains features from either $\mathbf{F}^{\text{3D}}$ or $\mathbf{F}^{\text{sph}}$. In addition, to enable features to interact directly with the global representation, we add two extra node features that are processed by the global average pooling module. The fusion graph node features are collected in the following set:

$$\mathbf{F}^{\text{merge}} = \left\{\mathbf{F}^{\text{3D}}, \mathbf{F}^{\text{sph}}, \text{Pooling}(\mathbf{F}^{\text{3D}}), \text{Pooling}(\mathbf{F}^{\text{sph}})\right\}. \tag{4.5}$$

Space-specific Interaction. We embed the fusion features $\mathbf{F}^{\text{merge}}$ into the Euclidean and hyperbolic spaces as $\mathbf{F}^{\mathbf{H}} = \exp_0^c(\mathbf{F}^{\text{merge}})$ (where the exp operator is applied node-wise) and $\mathbf{F}^{\mathbf{E}} = \mathbf{F}^{\text{merge}}$, respectively, to perform feature interaction using GA layers. Specifically, in the same way as (4.2), we obtain the k-th head FC features for $\mathbf{F}^{\mathbf{H}}$ or $\mathbf{F}^{\mathbf{E}}$, denoted as $\mathbf{F}_k^{\text{FC}}$. We additionally leverage a learnable matrix $\mathbf{M}$ regarded as a feature relationship metric such that the attention weights are computed as:

$$\mathbf{A}_k^{\mathbf{M}} = \text{Softmax}\Big(\mathbf{F}_k^{\text{FC}}\mathbf{M}\mathbf{F}_k^{\text{FC}^\intercal}\Big). \tag{4.6}$$

In Riemannian geometry, a Riemannian metric on a smooth manifold is a smooth symmetric covariant 2-tensor field that is positive definite at each point. The learnable matrix $\mathbf{M}$ can be viewed as a more general extension of the Riemannian metric, where we do not impose any constraint on it, leaving it to update freely. The effectiveness of this design can be seen in Table 4.5, where the free metric surpasses other counterparts.

The learned feature embeddings from the Euclidean and hyperbolic spaces are then passed into two different GA layers and finally fused together using element-wise adding:

$$\mathbf{F}^{\text{space}} = w^{\mathbf{E}}\mathbf{F}^{\mathbf{E}} + w^{\mathbf{H}}\mathbf{F}^{\mathbf{H}}, \tag{4.7}$$

where $w^{\mathbf{E}}$ and $w^{\mathbf{H}}$ denote learnable weights for the Euclidean and hyperbolic embeddings $\mathbf{F}^{\mathbf{E}}$ and $\mathbf{F}^{\mathbf{H}}$, respectively. Each node feature has thus aggregated information from both Euclidean and hyperbolic spaces, which can be viewed as an adaptive combination of linearity and non-linearity that can contribute to a more effective feature representation.

Modal-specific Interaction. We next decouple the merged features back to 3D features and projection features again, enabling them to turn around and learn information within their own modality. This is similar to the self-attention operation in the cross-self-attention pipeline. Specifically, for the 3D features, we pass them through a GA layer (with the learnable matrix) with preceding and succeeding MLP layers, while for the 2D projection features, we pass them through a basic ResNet block. After the modal-specific interaction, 3D features and projection features are merged together again using (4.5) to reconstruct the fusion features.

4.2.4 Pose Regression Head and Loss Function

The task of LiDAR pose regression requires predicting a 6-DoF pose. However, since the translation and rotation elements do not scale compatibly, the regression converges in different basins. To deal with this problem, previous methods [3, 38] consider the regression head with two parallel MLPs for translation and rotation regression, respectively. We thus use the same decoding head design as [3], which consists of two MLP layers for translation and rotation regression as shown in Fig. 4.3.

During training, to provide sufficient supervision to the whole pipeline, we use not only the fusion features but also the 3D and projection features at lower levels. As shown in Fig. 4.3, for the three features $\mathbf{F}^{\text{3D}}$, $\mathbf{F}^{\text{sph}}$, and $\mathbf{F}^{\text{merge}}$, we use three different regression heads $g^{\text{3D}}, g^{\text{sph}}, g$ respectively to predict their corresponding 6-DoF poses $(\mathbf{d}^{\text{3D}}, \mathbf{r}^{\text{3D}})$, $(\mathbf{d}^{\text{sph}}, \mathbf{r}^{\text{sph}})$, $(\mathbf{d}, \mathbf{r})$. Specifically, we first perform global average pooling

and then regression to obtain the predicted poses, which can be described as follows:

$$(\mathbf{d}^{\mathrm{3D}}, \mathbf{r}^{\mathrm{3D}}) = (g^{\mathrm{3D}} \circ \mathrm{Pooling})(\mathbf{F}^{\mathrm{3D}}), \tag{4.8}$$

$$(\mathbf{d}^{\mathrm{sph}}, \mathbf{r}^{\mathrm{sph}}) = (g^{\mathrm{sph}} \circ \mathrm{Pooling})(\mathbf{F}^{\mathrm{sph}}), \tag{4.9}$$

$$(\mathbf{d}, \mathbf{r}) = (g \circ \mathrm{Pooling})(\mathbf{F}^{\mathrm{merge}}), \tag{4.10}$$

where $\circ$ denotes the composition operation, and $\mathrm{Pooling}(\cdot)$ denotes the global average pooling operation. As for the rotation, we use the logarithmic format of the quaternion [3]. Denoting the translation and rotation targets as $\mathbf{d}^*$ and $\mathbf{r}^*$, the final loss function is computed as:

$$\begin{aligned}\mathcal{L} = &(\left\|\mathbf{d}^{\mathrm{3D}} - \mathbf{d}^*\right\| + \left\|\mathbf{d}^{\mathrm{sph}} - \mathbf{d}^*\right\| + \|\mathbf{d} - \mathbf{d}^*\|)e^{-\lambda} + \lambda \\ &+ (\left\|\mathbf{r}^{\mathrm{3D}} - \mathbf{r}^*\right\| + \left\|\mathbf{r}^{\mathrm{sph}} - \mathbf{r}^*\right\| + \|\mathbf{r} - \mathbf{r}^*\|)e^{-\rho} + \rho\end{aligned} \tag{4.11}$$

where λ and ρ are learnable parameters. During inference, the pose $(\mathbf{d}, \mathbf{r})$ is treated as the final prediction.

4.3 Experiments

In this section, we first evaluate our proposed model on datasets collected from outdoors and indoors. We next present ablation studies to demonstrate the effectiveness of our model design.

4.3.1 Implementation Details

We use ResNet34 [23] pre-trained on ImageNet as the backbone for projection features extraction. We use a batch size of 32. The number of attention heads is set as 8. Following [92], we set the base point $\mathbf{x}$ as 0 for hyperbolic embedding. We set $L^{\mathrm{3D}} = 2$ and $L = 2$. The Adam [109] optimizer with the initial learning rate 1×10^{-3} and weight decay 5×10^{-4} is used for training. We train our network for 150 epochs. All the experiments are conducted on either an NVIDIA RTX 3090 GPU or an NVIDIA RTX A5000 GPU.

	Model	Full-6	Full-7	Full-8	Full-9
LiDAR Retrieval	PointNetVLAD[4]	28.48 / 5.19	17.62 / 3.95	23.59 / 5.87	13.71 / 2.57
LiDAR Odometry	DCP[106]	18.45 / 2.08	14.84 / 2.17	16.39 / 2.26	13.60 / 1.86
Image-based PR	PoseLSTM[107]	26.36 / 6.54	74.00 / 9.85	128.25 / 18.59	19.12 / 3.05
	MapNet[28]	48.21 / 6.06	61.01 / 5.85	75.35 / 9.67	44.34 / 4.54
	AD-MapNet[29]	18.43 / 3.28	19.18 / 3.95	66.21 / 9.42	15.10 / 1.82
	AtLoc+[30]	17.92 / 4.73	34.03 / 4.01	71.51 / 9.91	10.53 / 1.97
	MS-Transformer[108]	11.69 / 5.66	65.38 / 9.01	88.63 / 19.80	7.62 / 2.53
	RobustLoc[19]	10.97 / 3.05	15.83 / 3.48	50.95 / 9.14	8.28 / 1.79
LiDAR-based PR	PointLoc[3]	13.81 / 1.53	9.81 / 1.27	11.51 / 1.34	9.51 / 1.07
	PosePN[38]	16.32 / 2.43	14.32 / 3.06	13.48 / 2.60	9.14 / 1.78
	PosePN++[38]	10.64 / 1.78	9.59 / 1.92	9.01 / 1.51	8.44 / 1.71
	PoseSOE[38]	8.81 / 2.04	7.59 / 1.94	9.21 / 2.12	7.27 / 1.87
	PoseMinkLoc[38]	11.20 / 2.62	14.69 / 2.90	12.35 / 2.46	10.06 / 2.15
	HypLiLoc (ours)	**6.00** / **1.31**	**6.88** / **1.09**	**5.82** / **0.97**	**3.45** / **0.84**

TABLE 4.1: Mean translation and rotation error (m/°) on the Oxford Radar dataset. The best and the second-best results in each metric are highlighted in **bold** and underlined, respectively. PR stands for pose regression. HypLiLoc achieves the best performance in all metrics.

	Model	Seq-05	Seq-06	Seq-07	Seq-14
Image-based PR	PoseLSTM[107]	0.16 / 4.23	0.18 / 5.28	0.24 / 7.05	0.13 / 4.81
	MapNet[28]	0.26 / 6.67	0.28 / 6.91	0.39 / 9.17	0.25 / 6.85
	AD-MapNet[29]	0.17 / 3.33	0.21 / 3.37	0.24 / 4.38	0.14 / 4.12
	AtLoc+[30]	0.18 / 4.32	0.24 / 5.14	0.26 / 6.04	0.16 / 4.61
	MS-Transformer[108]	0.16 / 3.98	0.15 / 3.56	0.18 / 5.32	0.13 / 4.83
	RobustLoc[19]	0.15 / 3.10	0.17 / 3.29	0.23 / 4.16	0.13 / 4.00
LiDAR-based PR	PointLoc[3]	0.12 / 3.00	0.10 / 2.97	**0.13** / 3.47	0.11 / 2.84
	PosePN[38]	0.12 / 4.38	0.09 / 3.16	0.17 / 3.94	**0.08** / 3.27
	PosePN++[38]	0.15 / 3.12	0.10 / 3.31	0.15 / 2.92	0.10 / 2.80
	PoseSOE[38]	0.14 / 3.15	0.11 / 2.90	0.15 / 3.06	0.11 / 3.20
	PoseMinkLoc[38]	0.16 / 5.17	0.11 / 3.74	0.21 / 5.74	0.12 / 3.64
	HypLiLoc (ours)	**0.09** / **2.52**	**0.08** / **2.58**	**0.13** / **2.55**	0.09 / **2.34**

TABLE 4.2: Median translation and rotation error (m/°) on the vReLoc dataset. The best and the second-best results in each metric are highlighted in **bold** and underlined, respectively. PR stands for pose regression. HypLiLoc achieves the best performance in 7 out of 8 metrics.

4.3.2 Datasets

4.3.2.1 Oxford Radar

The Oxford Radar dataset is a large-scale outdoor autonomous driving dataset [110]. It provides data from multi-modal sensors, including LiDARs, cameras, Radars, and GPS, but in our experiments, we use only LiDAR information. It contains sensor data in the time span of 1 year and a length span of 1000 km. In addition, it covers various seasons and weather conditions, which thus allows a comprehensive evaluation of the models. Following [3], we use the same benchmark data split

setting, and we also report the mean translation rotation error. The visualization is shown in Fig. 4.4.

4.3.2.2 vReLoc

The vReLoc dataset is an indoor robot dataset [3]. It consists of data from LiDARs, cameras, depth cameras, and motion trackers. In our experiments, we use only LiDAR information. It contains both static and dynamic scenarios with people walking around. Following [3], we use the same benchmark data split setting, and we also report the median translation and rotation error. The visualization is shown in Fig. 4.4.

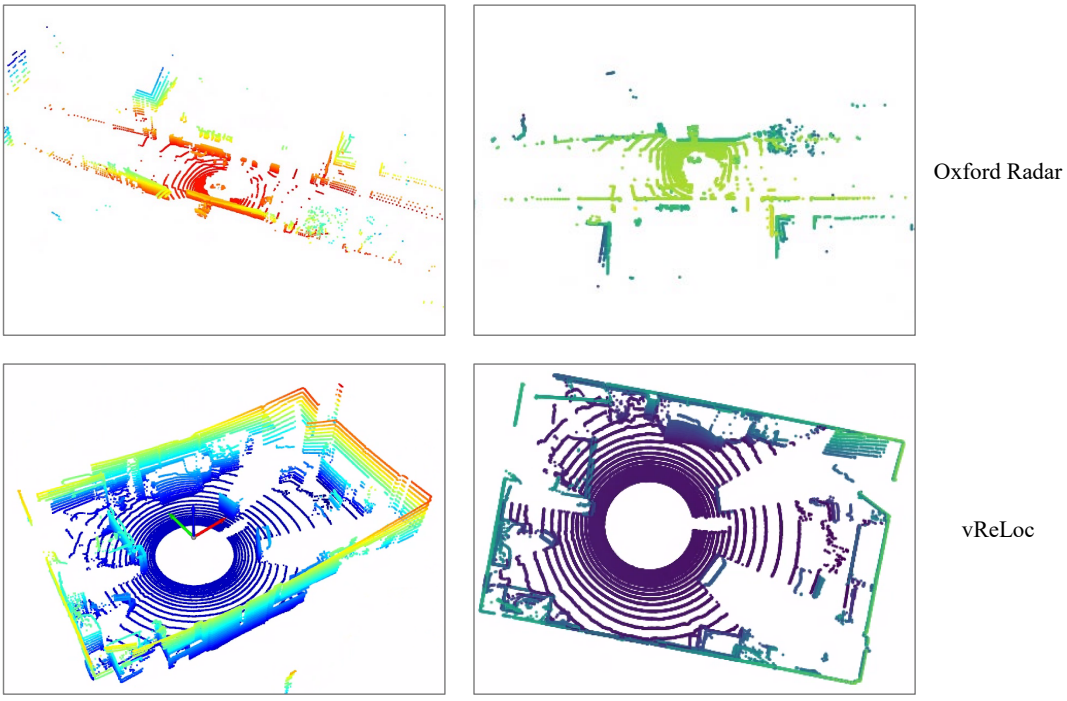

FIGURE 4.4: Visualization of the LiDAR point clouds used in the datasets.

4.3.3 Main Results

We first compare HypLiLoc with other baselines on the Oxford Radar dataset. From Table 4.1, we observe that HypLiLoc achieves SOTA performance in all metrics. Especially on the route Full-9, HypLiLoc obtains 3.45 m mean translation error in the city-wise localization task compared to the second best performer PoseSOE with

7.27 m, which demonstrates the effectiveness of HypLiLoc. In addition, compared with camera pose regression approaches that take images as inputs, LiDAR-based ones are generally more accurate. This verifies that point clouds generated by LiDARs are a more effective data modality for the localization task. Table 4.1 also indicates that for large-scale pose estimation, LiDAR pose regression approaches surpass both retrieval-based and odometry-based ones, and thus this approach is promising for many applications.

Note that pose regression approaches can be integrated into SLAM systems to achieve even better accuracy and to perform fast global pose estimating, especially in cases where a global navigation satellite system is not available (e.g., indoors and urban areas with dense skyscrapers).

We next test HypLiLoc on the indoor vReLoc dataset in Table 4.2, where it achieves SOTA performance in 7 out of 8 metrics and shows strong competitiveness. We note in the indoor environment, the LiDAR-based approaches also generally outperform image-based ones.

4.3.4 Ablation Studies

4.3.4.1 Module Ablation

We provide insights into our design choices for HypLiLoc by ablating each module. From Table 4.3, every module in our design contributes to the final improved estimation accuracy. Making use of information from the projected point cloud image and hyperbolic-Euclidean feature fusion strategy both contribute to more accurate pose regression outputs.

4.3.4.2 Different Projection Strategies

We next compare different modality strategies. As shown in Table 4.4, we first test the performance using the single modality input, including the 3D point cloud, the spherical projection, and the BEV projection. Among them, the 3D and the spherical projection show similar performances, while the BEV performance is worse. This verifies our insight that the BEV projection is not a bijective mapping, and thus less information is retained.

Method	Mean Error (m/°) on Full-8
base model	9.78 / 1.99
+ global graph attention	8.91 / 1.74
+ spherical-projection backbone	7.26 / 1.36
+ feature fusion block	6.19 / 1.13
+ learnable metric (full model)	**5.82** / **0.97**
full model w/o hyperbolic branch	6.57 / 1.19
full model w/o Euclidean branch	6.24 / 1.16

TABLE 4.3: Ablation study for different modules on Full-8 route of the Oxford Radar dataset.

#Modalities	3D	Sph.	BEV	Mean Error (m/°) on Full-8
1	✓			8.91 / 1.74
		✓		8.94 / 2.18
			✓	9.46 / 2.33
2	✓	✓		**5.82** / **0.97**
	✓		✓	9.44 / 1.80
		✓	✓	6.77 / 1.02
3	✓	✓	✓	6.32 / 1.01

TABLE 4.4: Comparison of different projection methods on Full-8 route of the Oxford Radar dataset. For the single modality, we do not use the feature fusion block.

When feeding two modalities, the combination *3D + spherical* surpasses the other two counterparts, which is our final model choice for HypLiLoc. When we further add the BEV input, the performance drops instead.

4.3.4.3 Learnable Matrix Design

We test the performance by applying different constraints on the learnable matrix $\mathbf{M}$ in (4.6). The Riemannian metric, which formulates a positive definite and symmetric matrix, has the strictest constraints. However, as shown in Table 4.5, the Riemannian metric does not provide performance improvements compared with the setting without any metric. If we only impose either the positive definite constraint or the symmetric constraint, the performance improves. Furthermore, if we exclude all constraints and enable the metric to evolve freely, we can achieve optimal performance.

Method	Mean Error (m/°) on Full-8
w/o **M**	6.19 / 1.13
Riemannian	6.34 / 1.28
positive definite	6.18 / 1.13
symmetric	6.02 / 1.24
no constraint	**5.82** / **0.97**

TABLE 4.5: Comparison of different constraints on Full-8 route of the Oxford Radar dataset.

	Model	Runtime Speed	Runtime Total Memory
Retrieval	PointNetVLAD[4]	11FPS	26GB
Odometry	DCP[106]	10FPS	22GB
Regression	PointLoc[3]	22FPS	7GB
	HypLiLoc (ours)	48FPS	6GB

TABLE 4.6: Comparison of the runtime speed and the runtime total memory of different models.

4.3.4.4 Computational Time and Storage

We compare LiDAR-based models that belong to different localization pipelines. As observed in Table 4.6, regression-based models can operate at least 2 times as fast as both the retrieval-based and odometry-based approaches. For the runtime memory, regression-based models need only 1/3 that of retrieval and odometry methods (less than 7 GB). HypLiLoc can perform inference at a speed of 48 FPS, which is 4 times as fast as the retrieval model and over 2 times as fast as PointLoc.

4.3.4.5 Visualization

We visualize a typical output pose trajectory of HypLiLoc and PointLoc in Fig. 4.5. HypLiLoc outputs a smoother and more accurate pose trajectory compared with PointLoc.

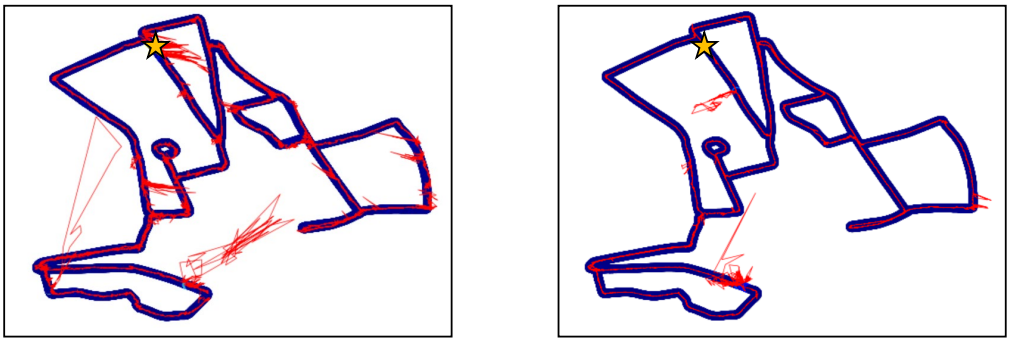

FIGURE 4.5: Trajectory visualization on the Oxford Radar dataset. The ground truth trajectories are shown in bold blue lines, and the estimated trajectories are shown in thin red lines.

4.4 Limitations

Although we have tested the proposed model in a city-wise dataset, verification of HypLiLoc's performance in challenging scenarios (e.g. with noise perturbations and adversarial attacks) is necessary for practical implementations.

4.5 Conclusion

In this work, we propose HypLiLoc, a novel network for LiDAR-based pose regression. It achieves effective feature extraction with global graph attention, hyperbolic-Euclidean interaction, and modal-specific learning. It achieves SOTA performance in both outdoor and indoor datasets.

Chapter 5

Multi-Modal Localization

In Chapters 3 and 4, we have introduced two single-modal pose regression networks. However, single-modal sensors always have limitations for conducting robust and effective localization in various environments. By contrast, a promising direction for enhancing localization performance is the integration of multiple visual sensors. Cameras provide dense, high-resolution images rich in texture and color, while LiDARs offer accurate 3D spatial measurements in the form of sparse point clouds. These sensing modalities are highly complementary: the semantic richness of images can enhance the geometric understanding of LiDAR.

Nevertheless, effectively fusing information from such heterogeneous data sources remains a major challenge. The fundamental differences in modality, structured 2D grids versus unstructured 3D point sets, make cross-modal alignment non-trivial. Achieving effective information fusion between image features and point cloud data is a critical step toward robust multi-modal visual localization.

Most current approaches [8, 66] focus primarily on global feature fusion and ignore the fine-grained local details that are essential for camera-LiDAR interaction. Moreover, they tend to overlook the valuable camera-LiDAR extrinsic calibration information, which could serve as a crucial bridge for better interaction between the two sensors.

To address these challenges, we propose the use of a learnable manifold metric attention mechanism that establishes a flexible fusion space, allowing for a more effective integration of 2D image and 3D LiDAR features. This fusion strategy

not only enables global feature interaction but also considers local relationships to enhance the model's understanding of the scene. Moreover, by incorporating camera-LiDAR extrinsic calibration, we extend the feature interaction to the pixel-point level, facilitating more coherent fusion between the different sensor modalities. In addition, we strengthen the feature extraction in the 2D branch by employing the ODE-based GNN. This integration enhances the model's robustness and improves its ability to extract meaningful features from visual data, especially in challenging environments. Ultimately, our proposed framework offers a more comprehensive solution for multi-modal visual localization.

5.1 Introduction

Cameras capture dense images and provide valuable visual cues, while LiDARs generate sparse point clouds that encode spatial geometry. The complementary nature of these sensors offers significant potential for more accurate and robust localization. However, the fundamental disparity between 2D image data and 3D spatial data presents a challenge for multi-modal feature fusion.

Existing multi-modal place recognition approaches such as MinkLoc++ [8] and AdaFusion [66] primarily focus on global feature fusion by pooling high-level descriptors from both modalities. Although this allows for coarse alignment of the sensor data, it tends to ignore the fine-grained local details that are crucial for precise camera-LiDAR interaction, particularly in complex environments where local geometry and visual details play an essential role in localization. Moreover, these approaches often overlook the camera-LiDAR extrinsic calibration, which can serve as a critical link for aligning features between the two sensors. The extrinsic calibration provides geometric constraints that could bridge the gap between 2D and 3D data, yet it remains underexplored in most existing fusion strategies.

To tackle these challenges, we introduce a learnable manifold metric attention mechanism that establishes a flexible fusion framework, facilitating more efficient integration of 2D image data and 3D LiDAR information. The manifold with metrics as in Chapter 2 provides a more flexible platform for multi-modal information fusion, thus helping build more effective localization features. This manifold-based strategy supports interaction at both global and local levels, which is essential for detailed

scene geometry comprehension. The adaptive nature of the manifold metric allows for dynamic fusion.

Furthermore, by incorporating camera-LiDAR extrinsic calibration into the fusion process, we extend the interaction from global alignment to a pixel-point level. This enables more precise correspondence between individual pixels in the 2D image and points in the 3D LiDAR cloud. The result is a more accurate and geometry-aware multi-modal representation that leverages both sensors.

In addition, we enhance feature extraction in the 2D camera branch by employing neural ODE-based GNNs. This integration improves the model's robustness, particularly in challenging environments with dynamic lighting, reflections, and other environmental perturbations. Neural ODEs provide the ability to handle noisy inputs, while GNNs allow for better aggregation of information from neighboring frames or sensors.

Ultimately, our proposed framework combines global and local feature fusion with robust and geometry-aware sensor integration. This comprehensive approach enables more precise visual localization by effectively utilizing both camera and LiDAR data, overcoming the limitations of previous methods that rely solely on global feature pooling. The fusion of multi-modal features at both global and pixel-point levels enhances the model's overall scene understanding ability to achieve more accurate localization in complex and dynamic environments.

The pipeline of our model is shown in Fig. 5.1. Our contributions are summarized as follows:

- We propose PRFusion, which leverages the Global Fusion Module (GFM) to enable detailed global multi-modal information interaction and integrates metric attention to achieve effective feature fusion. Additionally, PRFusion is equipped with the Neural Diffusion Module (NDM) to enhance feature robustness.
- We further introduce PRFusion++, which achieves pixel-point level multi-modal feature interaction by incorporating the Local Fusion Module (LFM) with camera-LiDAR extrinsic calibration information.

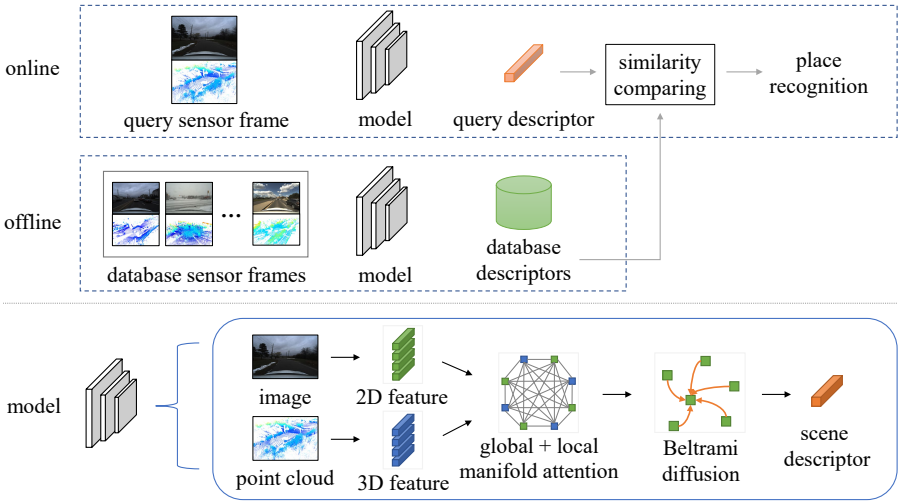

FIGURE 5.1: Our multi-modal place recognition pipeline. The query place is recognized by computing a scene descriptor based on both 2D and 3D features using our proposed place recognition model and then comparing it with the database descriptors. Our proposed model consists of global local feature fusion and neural Beltrami diffusion.

5.2 Methodology

In this section, we provide details of the model PRFusion, which does not need camera-LiDAR extrinsic information. As depicted in Fig. 5.2, given a sensor frame, we initially employ two distinct preliminary backbone networks to separately extract 2D and 3D feature maps, denoted as $\mathbf{F}^{2D} \in \mathbb{R}^{HW \times C}$ and $\mathbf{F}^{3D} \in \mathbb{R}^{N \times C}$, respectively. Here, H and W represent the height and width of the 2D feature map, N is the number of 3D features, and C is the dimension of the feature[1]. These different modal features are subsequently fed into the ensuing fusion blocks to facilitate feature interaction.

5.2.1 Global Fusion Module (GFM)

5.2.1.1 Fusion Feature Initialization

For multi-modal interaction, a pipeline that can effectively exploit information from distinct modalities is pivotal to the construction of the final scene descriptor. Nonetheless, previous multi-modal place recognition works [8, 66] only consider

[1]For illustration simplicity, we assume 2D and 3D features have the same dimension C.

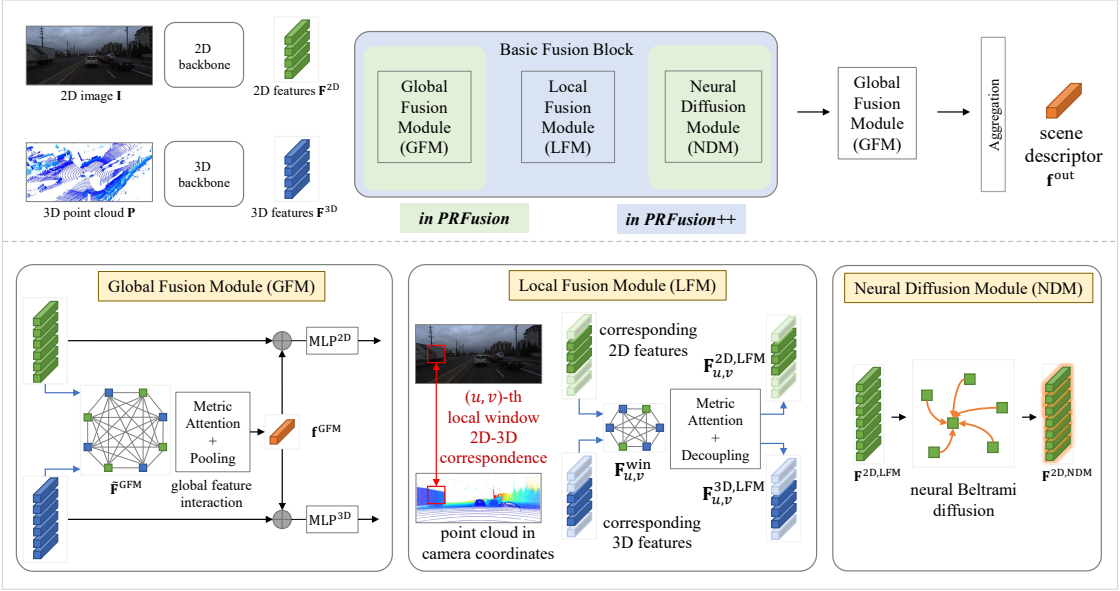

FIGURE 5.2: The overall architecture of our proposed PRFusion and PRFusion++. The multi-modal fusion is conducted in both the GFM and the LFM. The image features are additionally passed through the NDM to enhance the feature robustness.

fusing multi-modal features at the feature descriptor level, with interaction being applied only after the pooling operation. Such a late-stage processing strategy inherently overlooks a substantial amount of detailed information, making the multi-modal interaction less effective. To address this limitation, we introduce the GFM to achieve comprehensive 2D and 3D scene understanding.

To facilitate global multi-modal learning and generate a summarized vector to guide subsequent features, we construct a global multi-modal complete graph that comprises both 2D and 3D features. Given the significantly larger number of features, directly incorporating the entire features would impose a computational burden. Moreover, each feature, post backbone feature extraction, has already captured essential contextual information, thereby adequately representing the whole scene. Consequently, we opt for uniformly downsampled 2D and 3D features $\mathbf{F}^{2D\prime} \in \mathbb{R}^{N^{2D} \times C}$, $\mathbf{F}^{3D\prime} \in \mathbb{R}^{N^{3D} \times C}$ with highly summarized pooling features to expedite multi-modal interaction. Specifically, the initial graph features $\tilde{\mathbf{F}}^{\mathrm{GFM}} \in \mathbb{R}^{\tilde{N} \times C}$ with $\tilde{N} = N^{2D} + N^{3D} + 2$ can be obtained as:

$$\tilde{\mathbf{F}}^{\mathrm{GFM}} = \left\{ \mathbf{F}^{2D\prime}, \mathbf{F}^{3D\prime}, \mathrm{Pooling}\left(\mathbf{F}^{2D}\right), \mathrm{Pooling}\left(\mathbf{F}^{3D}\right) \right\}, \tag{5.1}$$

where Pooling(·) is the global average pooling operation.

5.2.1.2 Manifold Metric Attention

The attention mechanism has demonstrated its strong capability in various fields [25, 111]. In multi-modal learning, a significant challenge arises from the diverse nature of feature sources and their embedding within different feature spaces. A static fusion space may be insufficient to accommodate the increased diversity of these features. To achieve more effective feature interaction, we propose to endow the multi-modal fusion space with an adaptable similarity measurement.

The multi-modal features, denoted as $\mathbf{p}^{\mathcal{M}} = \tilde{\mathbf{F}}^{\mathrm{GFM}}$, are perceived as distinct base points within a manifold. These features construct their unique metric through a learnable neural ODE mapping defined as:

$$\frac{d\mathbf{g}_i(t)}{dt} = \sigma(\mathbf{g}_i(t)\mathbf{W}_{\mathbf{g}}), \tag{5.2}$$

where $\mathbf{g}_i(0) = \tilde{\mathbf{F}}_i^{\mathrm{GFM}}$, $\sigma(\,\cdot\,)$ is an non-linear activation function, and $\mathbf{W}_{\mathbf{g}} \in \mathbb{R}^{c\times C}$ is a learnable matrix. By solving (5.2), we obtain the diffused manifold metric at each base point as $\mathbf{G}_i$. Neural ODEs provide a flexible and adaptive framework for learning representations and are able to capture high-order dependencies, which supports constructing a more adaptive manifold metric and contributes to more fine-grained feature interaction of different modalities.

Each of the weighted global fusion features $\mathbf{F}_i^{\mathrm{GFM}} \in \mathbb{R}^C$ is then obtained as follows:

$$\mathbf{F}_i^{\mathrm{GFM}} = \sum_j a_{i,j}\tilde{\mathbf{F}}_j^{\mathrm{GFM}}\mathbf{W}_{\mathbf{V}}, \tag{5.3}$$

$$a_{i,j} = \mathrm{Softmax}\left(\left\langle \tilde{\mathbf{F}}_i^{\mathrm{GFM}}\mathbf{W}_{\mathbf{Q}}, \tilde{\mathbf{F}}_j^{\mathrm{GFM}}\mathbf{W}_{\mathbf{K}} \right\rangle_{\mathrm{diag}(\mathbf{G}_i)}\right), \tag{5.4}$$

where Softmax(·) denotes the row-wise softmax operation, $\mathbf{W}_{\mathbf{Q}}, \mathbf{W}_{\mathbf{K}}, \mathbf{W}_{\mathbf{V}}$ are learnable matrices, and diag(·) is the vector-to-matrix diagonal operation.

5.2.1.3 Feature Updating

The global representation of the scene can guide the 2D and 3D modality features towards better representation learning. To this end, we aggregate the whole global feature as a highly condensed feature vector $\mathbf{f}^{\mathrm{GFM}} \in \mathbb{R}^C$ summarized by the GeM pooling [43], $f_{\mathrm{GeM}}(\cdot)$, with a learnable parameter p defined as:

$$\mathbf{f}^{\mathrm{GFM}} = f_{\mathrm{GeM},p}\big(\mathbf{F}^{\mathrm{GFM}}\big) = \left(\frac{1}{\tilde{N}}\sum_{i=1}^{\tilde{N}}\big(\mathbf{F}_i^{\mathrm{GFM}}\big)^p\right)^{\frac{1}{p}}, \tag{5.5}$$

where $(\cdot)^p$ denotes raising each element to the power of p. We employ this vector as an element-wise additive bias to compute the biased 2D and 3D feature maps, enabling the respective modality features to encode holistic scene representations. The biased feature maps are then passed through separate MLPs to obtain the respective updated 2D or 3D feature maps, denoted as:

$$\mathbf{F}^{\mathrm{2D,GFM}} = \mathrm{MLP}^{\mathrm{2D}}\big(\mathbf{F}^{\mathrm{2D}} \oplus \mathbf{f}^{\mathrm{GFM}}\big) \in \mathbb{R}^{HW\times C}, \tag{5.6}$$

$$\mathbf{F}^{\mathrm{3D,GFM}} = \mathrm{MLP}^{\mathrm{3D}}\big(\mathbf{F}^{\mathrm{3D}} \oplus \mathbf{f}^{\mathrm{GFM}}\big) \in \mathbb{R}^{N\times C}, \tag{5.7}$$

where $\oplus$ denotes broadcast addition.

5.2.2 Neural Diffusion Module (NDM)

In challenging scenes where there are environmental perturbations, the sensors can be easily affected, leading to inferior feature extraction and poor place recognition performance. Image-based models are more susceptible to challenging environmental conditions than point cloud-based ones (see Table 5.4). Therefore, to achieve more robust feature expression, we resort to leveraging neural diffusion layers, where the outputs $\mathbf{F}^{\mathrm{2D,GFM}}$ from the GFM are additionally passed into the NDM for feature enhancement.

Specifically, we employ a neural diffusion mechanism rooted in the Beltrami flow. This mechanism is applied to each feature denoted as $\mathbf{X}(t)$, with $\mathbf{X}(0) = \mathbf{F}^{\mathrm{2D,GFM}}$.

The neural diffusion module is characterized as:

$$\frac{\mathrm{d}\mathbf{X}(t)}{\mathrm{d}t} = \overline{A}\mathbf{X}(t)\mathbf{W}_{\mathbf{X}} = (\mathbf{A}(t) - \mathbf{E})\mathbf{X}(t)\mathbf{W}_{\mathbf{X}}, \tag{5.8}$$

$$\mathbf{A}(t) = \mathrm{Softmax}(\mathrm{KNN}(\mathbf{Y}(t)\mathbf{Y}^{\intercal}(t))), \tag{5.9}$$

$$\mathbf{Y}(t) = \mathbf{X}(t)\mathbf{W}_{\mathbf{Y}}, \tag{5.10}$$

where $\mathbf{E}$ is the identity matrix, $\mathrm{KNN}(\cdot)$ is the K-nearest neighbor algorithm, $\mathbf{W}_{\mathbf{X}}, \mathbf{W}_{\mathbf{Y}}$ are learnable parameters. The neural Beltrami diffusion can be regarded as a type of graph neural diffusion, which is additionally equipped with graph topology rewiring achieved by positional embeddings. Graph rewiring enables the construction of flexible graph node connections and can thus contribute to better node feature learning.

By solving the above neural Beltrami diffusion equations, we obtain the 2D Beltrami features as $\mathbf{F}^{\mathrm{2D,NDM}}$. Subsequently, the 2D output $\mathbf{F}^{\mathrm{2D,NDM}}$ from the NDM and the 3D output $\mathbf{F}^{\mathrm{3D,GFM}}$ from the last GFM are together fed into the final GFM to achieve cascaded feature updating.

5.2.3 Output Scene Descriptor Generation

In the PRFusion framework, as shown in Fig. 5.2, we stack the above-described GFM for 2 layers with 1 intermediate NDM layer to achieve cascaded feature fusion. The final scene descriptor is obtained by concatenating the fusion, 2D, and 3D feature vectors obtained from the final GFM in (5.5) to (5.7):

$$\mathbf{e}^{\mathrm{out}} = \mathbf{f}^{\mathrm{GFM}} \,\big\|\, f_{\mathrm{GeM},p^{\mathrm{2D}}}\big(\mathbf{F}^{\mathrm{2D,GFM}}\big) \,\big\|\, f_{\mathrm{GeM},p^{\mathrm{3D}}}\big(\mathbf{F}^{\mathrm{3D,GFM}}\big), \tag{5.11}$$

where p^{2D} and p^{3D} are learnable parameters for the GeM pooling as in (5.5).

5.3 PRFusion++

The PRFusion model, as described above, is designed for multi-modal place recognition in scenarios where the extrinsic parameters between the camera and LiDAR sensors are unknown. Without knowing the extrinsic calibration, one cannot establish the correspondences between image pixels and point clouds. However, in most

applications, the sensor extrinsic parameters can be determined during the sensor setup stage. With this additional information, we can conduct multi-modal feature interaction at a more fine-grained level. Consequently, we propose an enhanced version of PRFusion, namely PRFusion++, which can leverage detailed 2D/3D information, as depicted in Fig. 5.2.

5.3.1 Local Fusion Module (LFM)

The projected points $\mathbf{I}^{\mathrm{3D}} = [\mathbf{u}^{\mathrm{3D}}, \mathbf{v}^{\mathrm{3D}}] \in \mathbb{R}^{N\times 2}$ on the image feature plane, derived from 3D cloud points, are given by the camera-LiDAR projection geometry.

Similarly, we can denote the image feature plane positions of the output 2D pixels as $\mathbf{I}^{\mathrm{2D}} = [\mathbf{u}^{\mathrm{2D}}, \mathbf{v}^{\mathrm{2D}}] \in \mathbb{R}^{HW\times 2}$.

The point cloud projection geometry enables an association between 2D and 3D features on the image plane, as illustrated in Fig. 5.2. This association facilitates the definition of a neighboring relation between 2D and 3D feature nodes. Hence, we introduce the LFM to construct local corresponding feature learning. Specifically, we divide the entire image feature frame into distinct non-overlapping windows [26], each with a size of $\Delta H \times \Delta W$. Each (u, v)-th window graph can contain a varying number of 2D and 3D feature nodes:

$$\begin{aligned}\mathbf{F}_{u,v}^{\mathrm{win}} = &\left\{\mathbf{F}_{j}^{\mathrm{2D,GFM}} \ : \ \left\lfloor \frac{\mathbf{I}_{j}^{\mathrm{2D}}}{[\Delta H, \Delta W]} \right\rfloor = [u, v]\right\}_{j\in[HW]} \\ &\bigcup \left\{\mathbf{F}_{j'}^{\mathrm{3D,GFM}} \ : \ \left\lfloor \frac{\mathbf{I}_{j'}^{\mathrm{3D}}}{[\Delta H, \Delta W]} \right\rfloor = [u, v]\right\}_{j'\in[N]},\end{aligned} \tag{5.12}$$

where $u \in \left\{1, \ldots, \lfloor \frac{H}{\Delta H} \rfloor\right\}$ and $v \in \left\{1, \ldots, \lfloor \frac{W}{\Delta W} \rfloor\right\}$. Within each window, we establish complete connections among all feature nodes, enabling each node to pass feature messages to all other nodes within the same window.

We then employ the manifold metric attention defined from (5.2) to (5.4) to perform local multi-modal feature updating for each window to obtain the updated features $\mathbf{F}_{u,v}^{\mathrm{LFM}}$ based on $\mathbf{F}_{u,v}^{\mathrm{win}}$. Note that the attention computation in all windows is proceeding in parallel with the assistance of the varied-length dot product package [111]. By decoupling the mixed feature nodes, $\mathbf{F}_{u,v}^{\mathrm{2D,LFM}}$ and $\mathbf{F}_{u,v}^{\mathrm{3D,LFM}}$, from each

window $\mathbf{F}_{u,v}^{\text{LFM}}$, we obtain the updated 2D and 3D features respectively as:

$$\mathbf{F}^{\text{2D,LFM}} = \Big\|_{u,v} \mathbf{F}_{u,v}^{\text{2D,LFM}}, \tag{5.13}$$

$$\mathbf{F}^{\text{3D,LFM}} = \Big\|_{u,v} \mathbf{F}_{u,v}^{\text{3D,LFM}}, \tag{5.14}$$

where the updated 2D features are fed into the NDM for robustness enhancement as shown in Fig. 5.2.

5.3.2 Output Scene Descriptor Generation

Different from PRFusion, besides 2 GFMs and 1 NDM, PRFusion++ is additionally equipped with 1 LFM as shown in Fig. 5.2. The final output scene descriptor is obtained using the outputs from the final GFM as in (5.11).

5.4 Loss Function

Both of our proposed models, PRFusion and PRFusion++, adopt the triplet loss [112]:

$$\mathcal{L} = \max\Big(\big\|\mathbf{e}^{\text{anchor}} - \mathbf{e}^{\text{positive}}\big\| - \big\|\mathbf{e}^{\text{anchor}} - \mathbf{e}^{\text{negative}}\big\| + m, 0\Big), \tag{5.15}$$

where $\mathbf{e}^{\text{anchor}}, \mathbf{e}^{\text{positive}}, \mathbf{e}^{\text{negative}}$ are descriptors of an anchor, a positive sample and a negative sample, and m is the margin hyperparameter.

5.5 Experiments

In this section, we compare our proposed models with other baselines in different datasets. We also conduct necessary ablation studies to demonstrate the effectiveness of our proposed modules.

5.5.1 Implementation Details and Datasets

5.5.1.1 Implementation Details

Our image and point cloud backbones are constructed based on MinkLoc++ [8]. We set the maximum batch size as 160. We train our network for a total of 120 epochs. The Adam optimizer [109] with a maximum learning rate $1e-3$ and weight decay $1e-4$ is used to train the network. We use SpTr [111] for varied-length multi-modal attention computing. The package torchdiffeq[75] is used for neural ODE solving. The input image is resized such that the short side is 240. The voxel quantization method is applied to input point clouds, where the quantization size is set as 0.1 for the Oxford dataset and 1 for the Boreas and KITTI datasets. Necessary data augmentation techniques are applied during training. The experiments are conducted on a Tesla A100.

5.5.1.2 Evaluation Metrics

We follow previous works to use the same evaluation protocol for place recognition, including Recall@1 (R@1), Average Recall@N (AR@N), and Average Recall@1% (AR@1%). Unless otherwise noted, we set the positive retrieval threshold as 25 m, i.e., if a retrieval is within 25 m from the query ground truth position, this retrieval is treated as a positive (successful) retrieval.

5.5.1.3 Datasets

The Oxford RobotCar dataset [101] is a large-scale autonomous driving dataset, including a wide range of driving conditions. We use the processed point clouds provided by PointNetVLAD[4] which is the standard benchmark data for point cloud place recognition. We test on both the standard benchmark split (denoted as Oxford-PNVLAD) and the split used by [65, 66] (denoted as Oxford-Cues). Since the processed point clouds break the camera-LiDAR extrinsic, we can only test PRFusion, which does not require the camera-LiDAR alignment.

We also test on the KITTI dataset[113], which is widely used in computer vision and autonomous driving research as it offers a diverse range of real-world autonomous

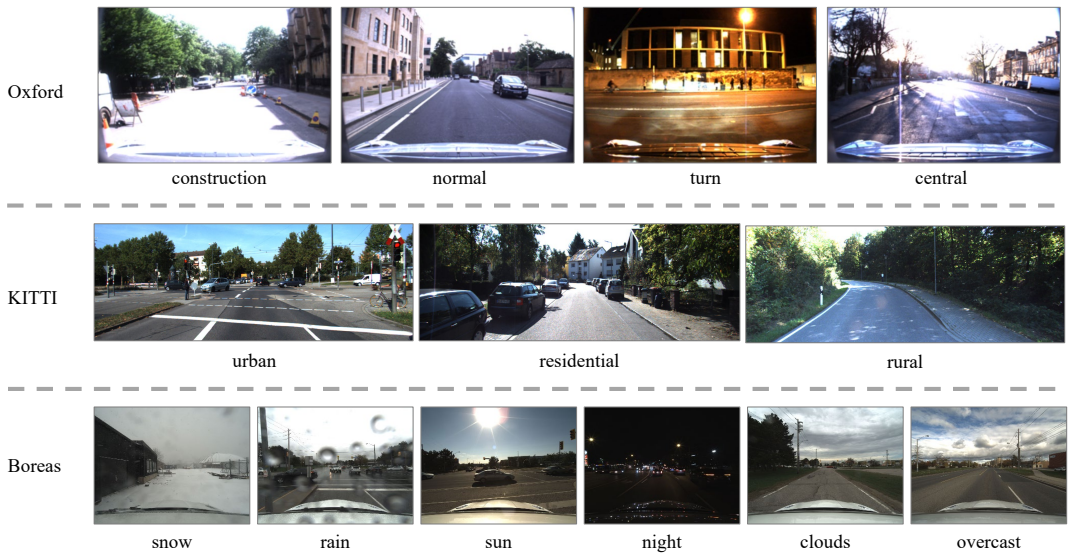

FIGURE 5.3: Examples from the Oxford, KITTI, and Boreas datasets.

vehicle data in different environments. It has provided camera-LiDAR extrinsic calibrations.

Both of the above two datasets are collected in normal conditions, which may not be sufficient to demonstrate the model's robustness. The Boreas dataset [114] is a more challenging dataset collected in various extreme conditions including snow, rain, and night. It also provides calibrated images and point clouds with synchronized extrinsic parameters.

Visual examples from the above datasets are provided in Fig. 5.3.

5.5.2 Main Results

We begin by comparing PRFusion with other baselines on the Oxford dataset. As shown in Tables 5.1 and 5.2, PRFusion achieves state-of-the-art performance in all metrics. This demonstrates the effectiveness of PRFusion. Furthermore, multi-modal place recognition approaches that take point clouds and images as inputs tend to be more accurate than those using only one modality. This verifies that images captured by cameras can provide additional useful information for the place recognition task.

We next conduct experiments on the KITTI dataset that encompasses various typical driving scenarios. As illustrated in Table 5.3, our proposed models consistently

	Model	Oxford-PNVLAD	
		AR@1%	AR@1
3D	PointNetVLAD [4]	80.3	63.3
	PCAN [52]	83.8	70.7
	LPD-Net [54]	94.9	86.4
	EPC-Net [55]	94.7	86.2
	SOE-Net [58]	96.4	89.3
	MinkLoc3D [56]	97.9	93.8
	MinkLoc3Dv2 [57]	98.9	96.3
	PTC-Net [115]	98.8	96.4
3D+RGB	CORAL [116]	96.1	-
	PIC-Net [64]	98.2	-
	MinkLoc++ [8]	99.1	96.7
	UMF[18]	<u>99.2</u>	97.2
	*UMF[18] w/ extra train data	99.1	<u>97.9</u>
	PRFusion (ours)	**99.6**	**98.2**

TABLE 5.1: place recognition results on the Oxford-PNVLAD dataset. All models do not use the re-ranking technique. "-" denotes that the result is not provided by the corresponding paper. "*" denotes that the model is trained with extra datasets. The best and the second best performances are marked with **bold** and <u>underline</u>, respectively.

	Model	Oxford-Cues	
		AR@1%	AR@1
3D+RGB	Cues-Net [65]	-	98.00
	AdaFusion [66]	<u>99.21</u>	<u>98.18</u>
	PRFusion (ours)	**99.94**	**99.08**

TABLE 5.2: place recognition results on the Oxford-Cues dataset. All models do not use the re-ranking technique or additional training datasets. "-" denotes that the result is not provided by the corresponding paper. The best and the second best performances are marked with **bold** and <u>underline</u>, respectively.

outperform other baselines across different evaluation metrics. The corresponding Average Recall@N curve, depicted in Fig. 5.4, demonstrates the superiority of our models.

We also visualize AR@1 under different positive retrieval thresholds (from 10 m to 25 m) as in Fig. 5.5. In challenging small thresholds ($<$ 16 m), comparable performance is observed with other baselines. However, under larger thresholds ($>$ 17 m), we achieve significantly better AR@1.

	Model	AR@1	AR@2	AR@5	AR@10
3D+RGB	MinkLoc++[8]	86.1	89.7	94.0	95.8
	AdaFusion[66]	86.7	90.6	93.9	96.1
	UMF[18]	86.3	89.9	94.2	96.3
	LCPR[67]	80.5	85.3	91.1	93.9
	PRFusion (ours)	<u>87.7</u>	**92.1**	<u>95.4</u>	<u>96.5</u>
	PRFusion++ (ours)	**88.7**	<u>91.7</u>	**95.8**	**97.2**

TABLE 5.3: place recognition results on the KITTI dataset. The best and the second best performances are marked with **bold** and <u>underline</u>, respectively.

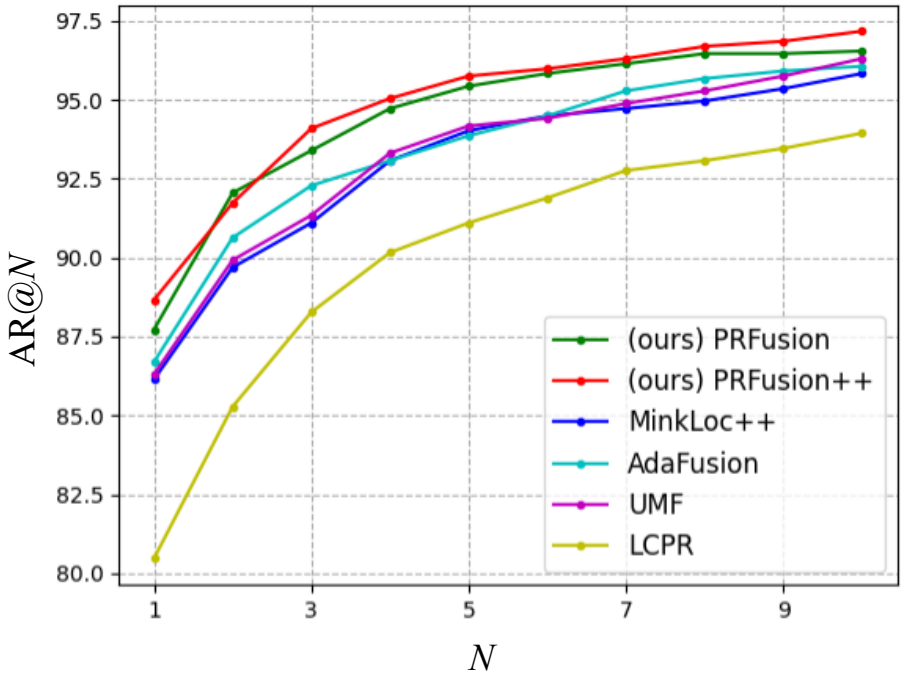

FIGURE 5.4: Average Recall@N curve on the KITTI dataset.

Next, we evaluate our models on the challenging Boreas dataset, which includes camera-LiDAR extrinsic calibrations and demonstrates the strong capacity of our advanced model, PRFusion++. As illustrated in Table 5.4, PRFusion++ achieves the highest score in all metrics. The superiority of leveraging multi-modal data is also evident in Table 5.4, as all multi-modal models outperform the single-modal ones. We also note that the point cloud models generally outperform the image models, particularly in the night scenario with extremely poor lighting conditions, where all image models exhibit significant performance drops while the point cloud models remain almost unchanged.

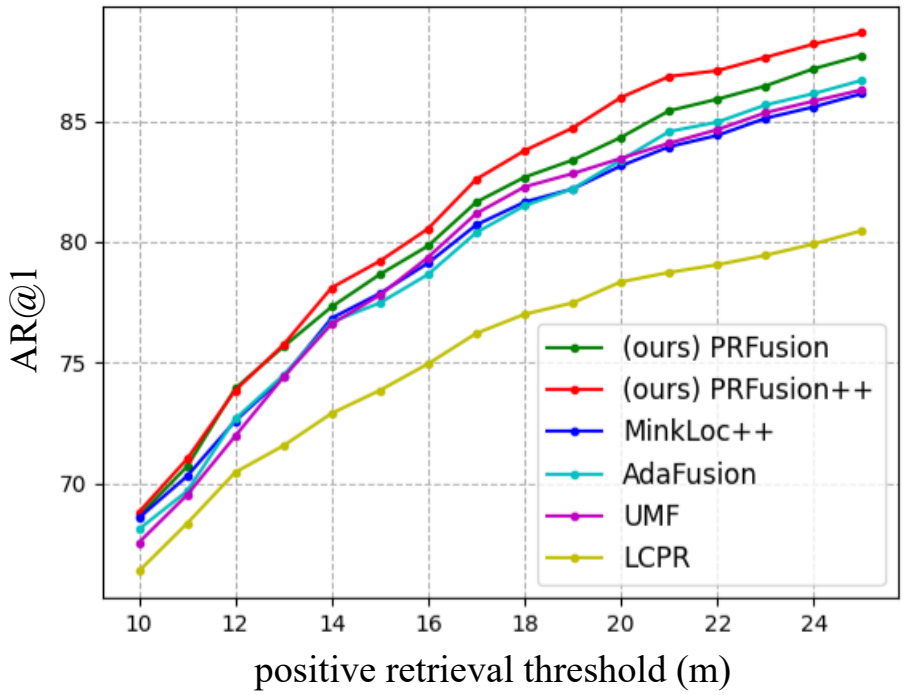

FIGURE 5.5: AR@1 under different positive retrieval thresholds on the KITTI dataset.

	Model	Snow	Rain	Sun	Night	Clouds	Overcast	AR@1 %	AR@1
		R@1	R@1	R@1	R@1	R@1	R@1		
RGB	GeM [43]	62.9	69.1	67.1	18.6	69.5	68.5	76.2	59.3
	ConvAP [44]	75.4	74.6	72.7	24.8	75.0	75.1	79.4	66.3
	MixVPR [45]	81.3	80.3	78.3	30.2	80.0	81.0	85.5	71.9
3D	MinkLoc3D [56]	77.0	89.5	90.0	89.3	88.6	89.6	96.6	87.3
	MinkLoc3Dv2 [57]	84.1	92.6	92.6	92.3	93.1	92.9	98.2	91.3
	PTC-Net [115]	82.2	91.4	92.2	92.0	92.8	93.1	97.4	90.6
3D + RGB	MinkLoc++ [8]	87.1	93.3	94.6	90.9	94.5	94.5	98.9	92.5
	AdaFusion [66]	88.1	93.5	94.8	91.5	94.8	94.7	99.0	92.9
	UMF [18]	87.5	94.2	94.0	91.4	93.8	93.4	99.0	92.4
	LCPR[67]	70.3	90.2	88.9	85.1	89.6	89.7	95.3	85.6
	PRFusion (ours)	<u>92.0</u>	<u>96.4</u>	<u>95.0</u>	<u>92.5</u>	<u>95.9</u>	<u>95.8</u>	<u>99.2</u>	<u>94.6</u>
	PRFusion++ (ours)	**94.2**	**96.7**	**96.1**	**95.9**	**96.1**	**96.3**	**99.6**	**95.9**

TABLE 5.4: place recognition results on the Boreas dataset. All models do not use the re-ranking technique or additional training datasets. The best and the second best performances are marked with **bold** and <u>underline</u>, respectively.

5.5.3 Design Analysis

To verify the effectiveness of the proposed modules, we conduct design analysis for PRFusion++ in the challenging Boreas dataset.

5.5.3.1 Module Ablation

We first ablate the proposed modules from PRFusion and PRFusion++ as shown in Table 5.5, where they all contribute to better performance. The three proposed modules (GFM, NDM, and LFM) together yield +3.4 gains for AR@1, underscoring their efficacy in enhancing overall outcomes. Among them, the GFM and LFM contribute more than the NDM.

Method	AR@1	Δ
PRFusion	**94.6**	-
PRFusion w/o global fusion module (GFM)	93.2	-1.4
PRFusion w/o neural diffusion module (NDM)	93.8	-0.8
w/o GFM/NDM	92.5	-2.1
PRFusion++	**95.9**	-
PRFusion++ w/o global fusion module (GFM)	94.7	-1.2
PRFusion++ w/o neural diffusion module (NDM)	95.3	-0.6
PRFusion++ w/o local fusion module (LFM)	94.6	-1.3
w/o GFM/NDM/LFM	92.5	-3.4

TABLE 5.5: Main ablation study on the proposed modules.

5.5.3.2 Manifold Metric Attention

In Table 5.6, we present a comparative analysis among MLP, the conventional vanilla attention mechanism, and our proposed metric attention approach. Our metric attention scheme yields a noteworthy improvement of +2.2 in the AR@1 metric. Compared to vanilla attention, which contributes +1.5, our metric attention demonstrates a +0.7 better performance. This indicates its efficacy in facilitating more robust multi-modal interactions through the acquired adaptive manifold metric.

Furthermore, we conduct ablation experiments on the metric attention module by selectively removing the neural ODE and the non-linear activation function. These ablations result in a noticeable performance degradation, underscoring the critical role played by the amalgamation of higher-order ODE modeling and non-linear activation functions in the construction of our flexible metric framework.

Method	AR@1	Δ
w/o attention (MLP)	93.7	-
vanilla attention	95.2	+1.5
metric attention w/o ODE	95.5	+1.8
metric attention w/o activation	95.6	+1.9
metric attention	**95.9**	**+2.2**

TABLE 5.6: Comparison of different types of vanilla attention and our proposed metric attention.

5.5.3.3 Sampling Points in the GFM

Within the GFM, we adopt a strategy of leveraging down-sampled 2D/3D features for the synthesis of multi-modal features. As in Table 5.7, with a modest number of sampled features, we are able to effectively encapsulate global-level representations. This empirical observation serves as compelling evidence that our design is highly effective and well-suited for its intended purpose.

#Points	AR@1
8	95.2
16	**95.9**
32	95.7
64	95.8

TABLE 5.7: Comparison on the number of sampled points in the GFM.

5.5.3.4 Fusion Window Size in the LFM

In the LFM, we perform 2D/3D feature interaction within a local window. Our experimentation, as outlined in Table 5.8, involves the exploration of varying window sizes. Intriguingly, the results clearly demonstrate that the smallest window size, 1×1, emerges as the optimal selection. This is attributed to its capacity to facilitate fine-grained local feature interactions. Conversely, with the increasing window size, the performance decreases steadily. This indicates that trivially conducting full-scale multi-modal feature interaction is not a suitable solution for feature updating.

$\Delta H \times \Delta W$	AR@1
1×1	**95.9**
2×2	95.2
4×4	94.3
8×8	92.1

TABLE 5.8: Comparison on different window size $\Delta H \times \Delta W$ in the LFM.

5.5.3.5 Neighborhood Scale in the NDM

The neural Beltrami diffusion in the NDM is based on adaptive neighborhood construction, and we test the performance with different neighbors in the NDM. As illustrated in Table 5.9, a medium scale with 25 neighbors is an optimal choice for Beltrami feature diffusion. Further bringing more neighbors can not contribute to better performance.

#Neighbors	AR@1
9	95.4
16	95.7
25	**95.9**
36	95.6

TABLE 5.9: Comparison on the number of nearest neighbors in the NDM.

5.5.3.6 Robustness Against Image Perturbations

To verify the model robustness in more challenging scenarios, we test to add additional Gaussian noise onto the original image $\mathbf{I}$. The perturbed image is built as $\hat{\mathbf{I}} = \mathbf{I} + \alpha\delta$, where $\delta \sim \mathcal{N}(0, 1)$ and $\alpha > 0$ controls the intensities, and the descriptor of the perturbed scene can be denoted as $\hat{\mathbf{e}}$. The visualization of perturbed images with different intensities is shown in Fig. 5.6. As plotted in Fig. 5.6, our proposed models PRFusion and PRFusion++ both show significantly better robustness against image perturbations, which strongly underscores the efficacy of our design. We also test the robustness by excluding the NDM, which leads to inferior performance in challenging scenarios. Moreover, it can be noticed from Fig. 5.6 that the LFM in PRFusion++ can boost model robustness for a large margin compared with PRFusion. In addition, we visualize the kernel density estimate plots and box plots as in Fig. 5.7. With the integration of the NDM, our model can lead to smaller

$\|\mathbf{e}^{\text{out}} - \hat{\mathbf{e}}^{\text{out}}\|$ compared with that without the NDM. The robustness of the NDM can thus be demonstrated.

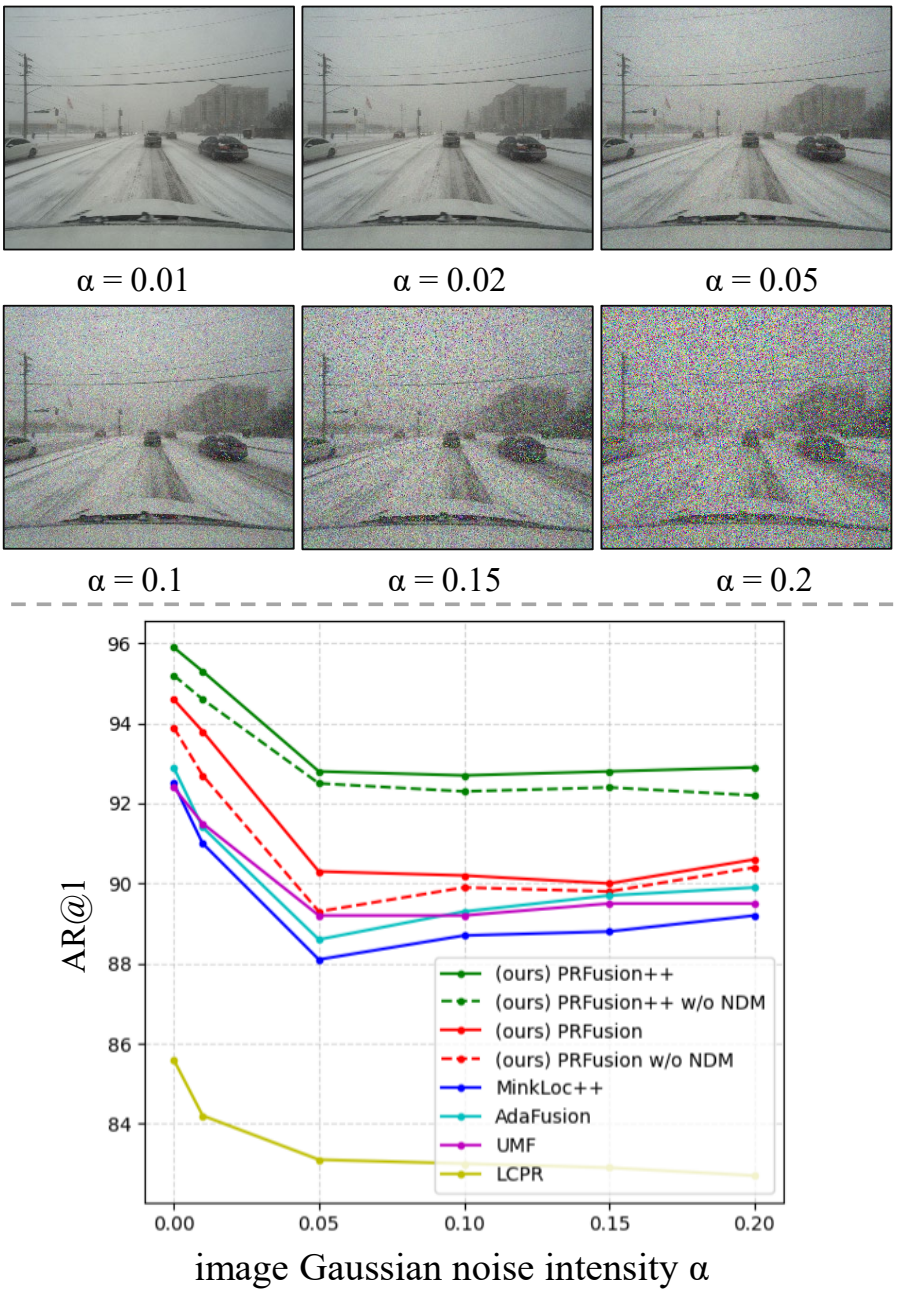

FIGURE 5.6: Above: visualization of images under additive Gaussian noise with different noise intensities α. Below: performance plot (AR@1) under additive image Gaussian noise with different noise intensities α.

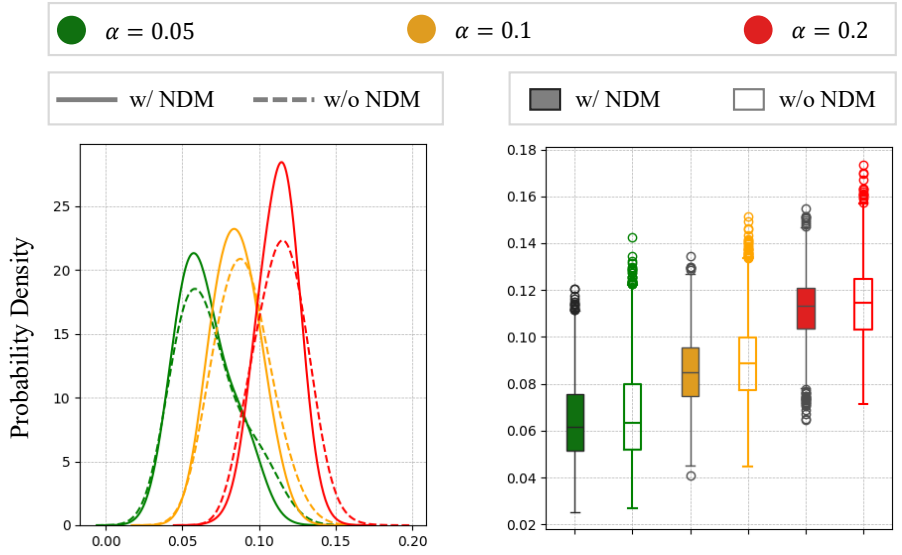

FIGURE 5.7: Left: kernel density estimate plots. Right: box plots of $\left\|\mathbf{e}^{\text{out}} - \hat{\mathbf{e}}^{\text{out}}\right\|$ under additive Gaussian noise with different noise intensities α.

5.5.3.7 Robustness Against Extrinsic Calibration Errors

Our tested Boreas dataset is not with a perfectly time-synchronized camera-LiDAR suite. As shown in Figure Fig. 5.8, due to the ego-vehicle's movement and imperfect calibration, significant misalignments occur as highlighted in red boxes. Our models can still operate well under such scenarios, which show the basic robustness against calibrations.

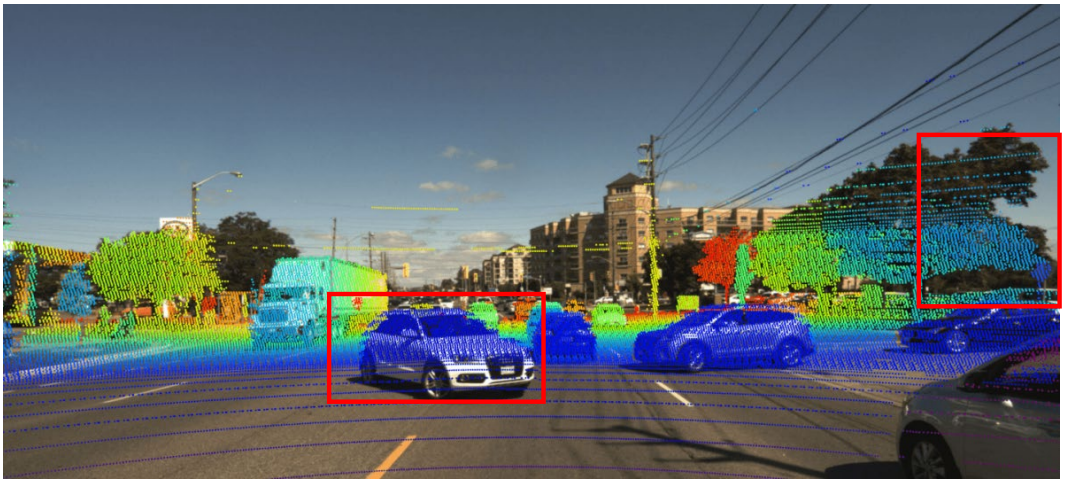

FIGURE 5.8: Visualization of the projected LiDAR point cloud onto the image in the Boreas dataset. Misalignments are highlighted in red boxes.

We evaluate the robustness against camera-LiDAR extrinsic calibration parameter errors. We denote the true calibration translation vector and calibration rotation matrix as $\mathbf{d}$ and $\mathbf{R}$ respectively. The measured calibration translation vector and rotation matrix can be denoted as $\hat{\mathbf{d}} = \mathbf{d} + \mathbf{d}_e$ and $\hat{\mathbf{R}} = \mathbf{R}_e\mathbf{R}$ with rotation matrix and translation vector errors $\mathbf{R}_e$ and $\mathbf{d}_e$. The calibration translation error t_e and rotation angle error r_e are given as:

$$
\begin{aligned}
d_e &= \|\mathbf{d}_e\|_2 \,, \\
r_e &= \left| \arccos\left(\frac{\text{trace}(\mathbf{R}_e) - 1}{2} \right) \right| .
\end{aligned}
$$

The visualization of different extrinsic calibration errors is shown in Fig. 5.9. The performance comparison under different error thresholds is shown in Fig. 5.9. The results demonstrate that reasonably good extrinsic calibration is essential for the PRFusion++ model as it utilizes the camera-LiDAR extrinsic calibration information. In practice, the extrinsic calibration error is typically less than 0.1 m and 1°[117–121]. From Fig. 5.9, we see that under such errors, PRFusion++ outperforms other baselines, demonstrating its effectiveness.

Model	Speed (FPS)	GPU Mem. (GB)
MinkLoc++	95	**2.0**
AdaFusion	84	**2.0**
UMF	90	**2.0**
LCPR	**110**	2.1
PRFusion (ours)	65	2.2
PRFusion++ (ours)	40	2.3

TABLE 5.10: Runtime speed and GPU memory usage on a Tesla A100.

5.5.3.8 Run Time Speed and GPU Memory Usage

Finally, we present the run time speed and GPU memory usage in Table 5.10. Compared with baselines, PRFusion and PRFusion++ are computationally costlier with the inclusion of the attention operation. However, these models still meet typical real-time deployment requirements (FPS > 30). The higher computational cost can be mitigated with various solutions, such as knowledge distillation, quantization, and pruning, which are interesting future work for further improvements.

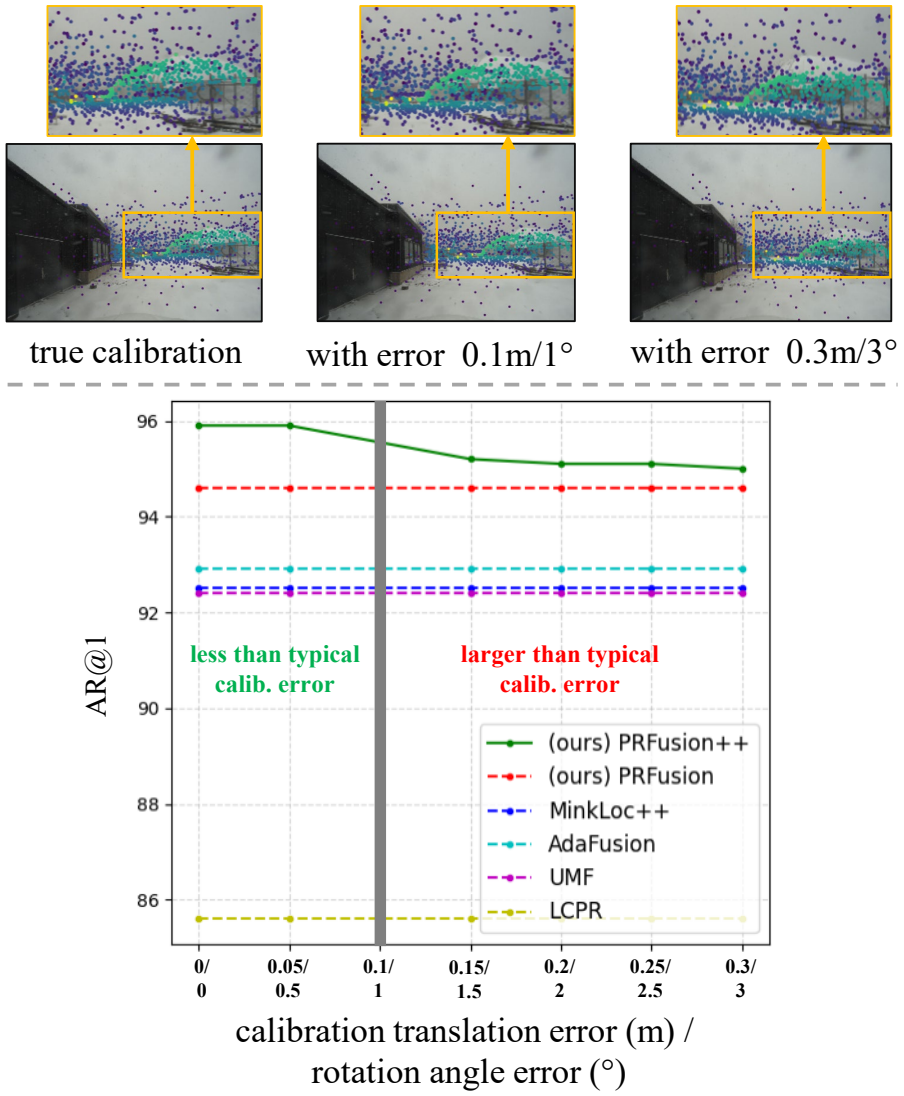

FIGURE 5.9: Above: visualization of images and point clouds under camera-LiDAR extrinsic parameter calibration errors. Below: performance plot (AR@1) under camera-LiDAR extrinsic parameter calibration errors. Note that different from PRFusion++, PRFusion and other baselines are not affected by extrinsic parameter calibration errors and their performance plots are flat.

5.6 Conclusion and Limitations

We have designed two multi-modal place recognition models, PRFusion and PRFusion++, which leverage manifold-based attention to facilitate more effective feature fusion at both global and local levels. Additionally, these models are equipped with neural Beltrami diffusion for more robust feature learning. The experimental results on large-scale benchmarks demonstrate the effectiveness of our design, as both models achieve SOTA performance.

However, due to separate feature extraction for 2D and 3D inputs and the additional computational overhead of attention computations, these models are computationally more expensive than several existing baselines. An interesting avenue for future work is to improve the pipeline for faster and lighter place recognition. To achieve this goal, we can explore various solutions, such as knowledge distillation, quantization, and pruning, which offer promising directions for further optimization.

Chapter 6

Single-Modal Localization Boosted by Multiple Modalities

In Chapter 5, we propose multi-modal place recognition networks that leverage information from both cameras and LiDARs. On the one hand, introducing multiple sensors can significantly boost localization performance by providing richer and more diverse data. On the other hand, this improvement comes at a cost. Adding extra sensors increases not only the hardware expenses but also the system's complexity. For mobile agents that require lightweight and efficient operation, incorporating too many sensors can reduce their flexibility and overall efficiency. A promising solution to this problem is KD, which enables knowledge transfer from a strong multi-modal teacher model (using multiple sensors) to a lightweight and single-modal student model. This approach allows us to maintain high performance while reducing the reliance on additional sensors.

However, most existing KD methods[70, 122, 123] are designed for tasks like classification and object detection, where geometric features and relationships are not sufficiently considered. In place recognition, the primary goal is to localize the query by matching it against a reference database. This matching process relies heavily on the geometric and spatial relationships between features, making it critical to ensure these relationships are preserved during knowledge transfer. Existing KD approaches overlook this important aspect, limiting their effectiveness for visual localization.

To address this gap, we propose an approach that goes beyond standard KD techniques by specifically focusing on the relationships between the teacher and student models. By exploring multiple types of relationships, we aim to better capture and transfer the matching knowledge. Additionally, our approach leverages spaces with varying curvatures, which allows for a more effective feature embedding. By computing distances in diverse geometric spaces, we can more flexibly reflect the geodesic distances between features, leading to a more effective and geo-consistent knowledge transfer. This method ensures that the spatial relationships are preserved and can thus be a powerful solution for lightweight and high-performance visual localization.

6.1 Introduction

As introduced in Chapter 5, integrating various sensors can elevate place recognition model performance, but it also incurs additional expenses. Moreover, lightweight mobile systems might not support heavy sensors, such as LiDARs, making multi-modal sensors impractical in some lightweight operation scenarios.

Although using multiple sensors during inference is not favored, we can harness this cross-modal knowledge during student model training. This is where cross-modal KD enters the picture. Specifically, in the student training phase, as depicted in Fig. 6.1, distinct modalities can be fed into the pre-trained teacher model. The extracted teacher features can then guide single-modal student models in learning superior features through additional supervision. During inference, student models can still rely on single-modal data, eliminating the need to accommodate multiple sensors.

Given the inconsistency in feature embedding across different modalities, directly compelling students to learn teacher features would be intricate. In contrast, the relational KD paradigm [71], which delves into feature relationships, offers a more suitable approach to address this inconsistency. However, the vanilla relational KD solution only considers feature relationships in limited embedding spaces, and they restrict relationship computing within the same knowledge agents (i.e. either teacher-teacher relationships or student-student relationships). These limitations hinder the efficient transfer of knowledge from teachers to students.

To mitigate these issues, we propose to extend the scope of feature relationships, encompassing both *self-agents* and *cross-agents* to facilitate a more comprehensive exploration of knowledge. In addition, our approach performs feature embedding in *multiple manifolds* with diverse feature geodesic measurements, enhancing the construction of effective feature relationships. Based on these designs, we propose DistilVPR as a cross-modal distillation pipeline for visual place recognition.

Through extensive experiments, we showcase the remarkable performance of DistilVPR when compared to previous KD baselines. Our approach achieves SOTA performance in the task of cross-modal distillation for visual place recognition. Furthermore, we rigorously investigate our design through vital ablation studies, providing empirical evidence of the efficacy of our proposed methodology.

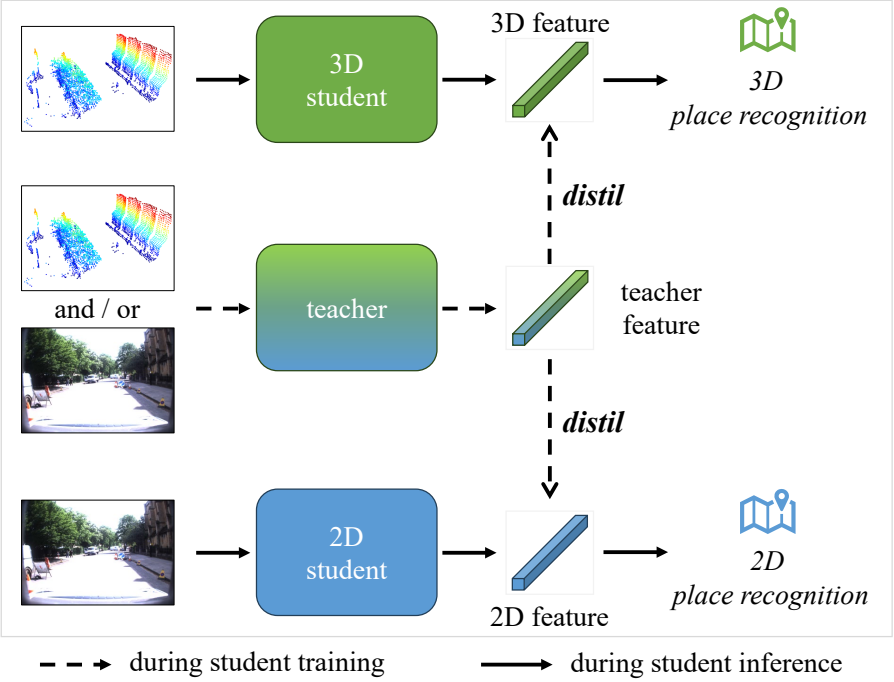

FIGURE 6.1: The pipeline of cross-modal KD to transfer knowledge from the cross-modal teacher to single-modal students.

6.2 Methodology

In this section, we first provide the problem formulation. Then, we introduce the DistilVPR architecture in detail.

6.2.1 Problem Formulation

In this study, we address the challenge of cross-modal KD for visual place recognition. We focus on a scenario where a pre-trained teacher model is provided, capable of processing images and/or point clouds as inputs for multi-modal visual place recognition. The single-modal student models accept either image inputs or point cloud inputs. Our objective is to distill the teacher's knowledge to the students, empowering them to acquire enhanced understanding during training. This, in turn, improves student performance during inference without the requirement for cross-modal sensors.

Specifically, we denote a batch of teacher outputs as[1] $\mathbf{T} = \{\mathbf{t}_i \in \mathbb{R}^C \,:\, i \in [B]\}$ and student outputs as $\mathbf{S} = \{\mathbf{s}_i \in \mathbb{R}^C \,:\, i \in [B]\}$, with the batch size B and the same output channel size[2] C .

6.2.2 Relational Distillation

There are typically two ways to conduct KD, including direct KD and relational KD. Direct KD is a straightforward way that directly applies sample-wise supervision by minimizing the loss

$$\mathcal{L}_{\text{direct}} = \sum_{i \in [B]} \ell(\mathbf{t}_i, \mathbf{s}_i), \tag{6.1}$$

where $\ell(\,\cdot\,)$ denotes the loss function. This approach pulls student embeddings towards teacher embeddings, which can be regarded as sample-wise supervision.

By contrast, relational KD does not apply explicit sample-wise supervision. Instead, it measures inter-sample relationships, which can be regarded as implicit knowledge. Relational KD is formed by minimizing

$$\mathcal{L}_{\text{relationship}} = \sum_{i,j \in [B]} \ell(r(\mathbf{t}_i, \mathbf{t}_j),\ r(\mathbf{s}_i, \mathbf{s}_j)\ ), \tag{6.2}$$

where $r(\,\cdot\,,\,\cdot\,)$ is the relational function to compute embedding distances.

[1] We denote $[B] = \{1, \ldots, B\}$ for simplification.
[2] We assume the teacher and the student have the same output channel size.

Through our experiments, we have observed that compared with direct KD, relational KD is inherently a better choice for cross-modal KD in visual place recognition for the following reasons. On one hand, in visual place recognition, places are recognized by computing query-database similarity, where the training goal is to minimize the query-positive distance and maximize the query-negative distance. The relative feature relationships are more critical than the absolute feature embeddings, for which the relational KD scheme that explores relative embedding distance would be a more suitable solution for visual place recognition. On the other hand, cross-modal features may have inherently different embedding patterns. Thus it would be intractable to force single-modal features to be embedded in the same space as multi-modal features using direct KD schemes. Based on these insights, we follow the relational KD scheme in (6.2) to design a more efficient cross-modal distillation solution.

6.2.3 Multi-agent Relationship

We generically call a teacher output $\mathbf{t}_i$ or student output $\mathbf{s}_i$ an *agent*. One limitation of the basic relational KD is that it confines the computation of relationships within the same type of agent, i.e., teacher-teacher $r(\mathbf{t}_i, \mathbf{t}_j)$ and student-student $r(\mathbf{s}_i, \mathbf{s}_j)$. Despite relational KD being able to achieve considerably better performance than direct KD counterparts, it lacks a more generalized consideration of the combination of different agents.

To generalize the combination of different agents, there are three scenarios for relationship computation as shown in Fig. 6.2. We expand (6.2) to further explore not only the self-agent relationships, $r(\mathbf{t}_i, \mathbf{t}_j)$ and $r(\mathbf{s}_i, \mathbf{s}_j)$, but also the cross-agent relationship, $r(\mathbf{t}_i, \mathbf{s}_j)$. Specifically, the three generalized relational KD losses are formulated as:

$$\mathcal{L}_{\text{tt}-\text{ss}} = \sum_{i,j\in[B]} \ell(r(\mathbf{t}_i, \mathbf{t}_j),\ r(\mathbf{s}_i, \mathbf{s}_j)\ ), \tag{6.3}$$

$$\mathcal{L}_{\text{ts}-\text{ss}} = \sum_{i,j\in[B]} \ell(r(\mathbf{t}_i, \mathbf{s}_j),\ r(\mathbf{s}_i, \mathbf{s}_j)\ ), \tag{6.4}$$

$$\mathcal{L}_{\text{tt}-\text{ts}} = \sum_{i,j\in[B]} \ell(r(\mathbf{t}_i, \mathbf{t}_j),\ r(\mathbf{t}_i, \mathbf{s}_j)\ ). \tag{6.5}$$

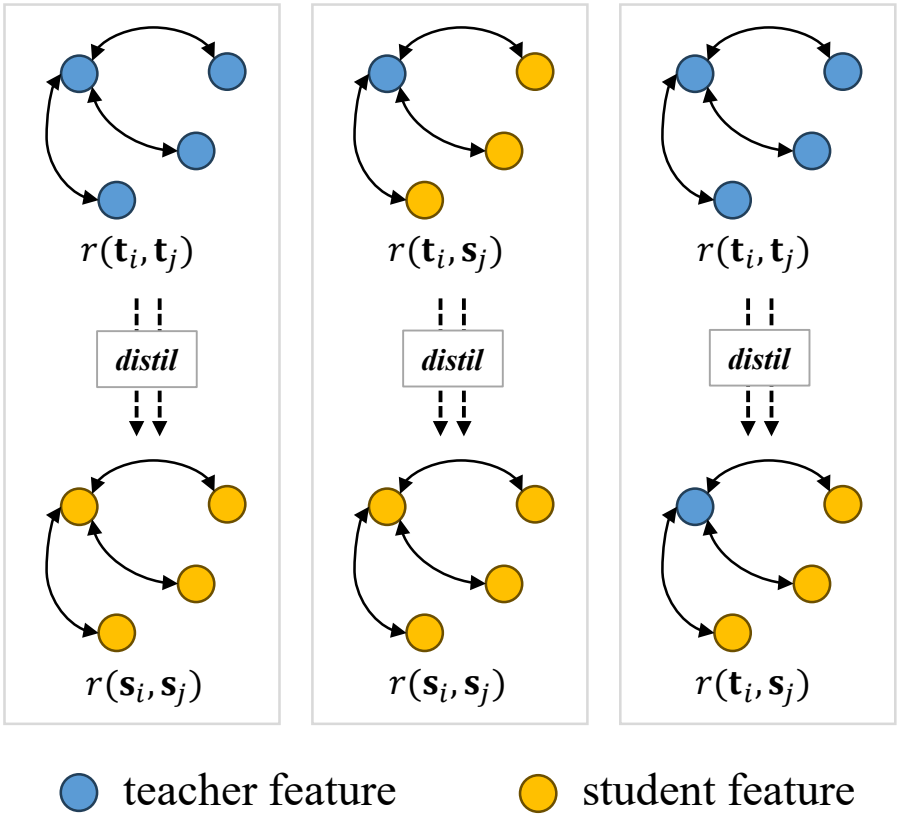

FIGURE 6.2: Three generalized relational KD schemes.

From above, (6.3) is the vanilla relational KD scheme in (6.2). By contrast, (6.4) and (6.5) are two additional schemes with (6.5) used in CSD [124]. In the two schemes, cross-agent relationship $r(\mathbf{t}_i, \mathbf{s}_j)$ bridges the gap between teacher features and student features. The additional information would contribute to a more effective KD process.

Comparing (6.4) and (6.5), we have empirically found that (6.4) generally outperforms (6.5), as seen in Table 6.4. This may be attributed to the fact that within the four variables in (6.5), only one pertains to the learnable student features, while the remaining three variables are associated with the fixed teacher features. Consequently, the optimal solution domain for minimizing (6.5) becomes constrained. For example, considering r as the Euclidean distance function, the optimal solution is confined to a sphere. Similarly, this constraint could potentially elucidate why direct KD in (6.1) yields inferior results compared to relational KD, given that direct KD also involves only one variable for student features, resulting in the optimal solution domain that is a single point.

Based on these insights, we thus use (6.4) to compute cross-agent relationships,

along with (6.3) for self-agent relationships. These distinct relationship patterns constitute the core components of our approach.

6.2.4 Multi-manifold Relationship

Since different modal features may not be embedded similarly, it becomes essential to adopt a more comprehensive metric for measuring agent relationships. Consequently, we introduce combined manifold spaces to augment the effectiveness of relational KD.

Different feature manifolds can be categorized based on their curvature. The Euclidean space represents the most prevalent manifold with zero curvature, while the spherical manifold exhibits positive curvature, and the hyperbolic manifold has negative curvature. By amalgamating multiple manifolds, we can facilitate features to possess more comprehensive embedding relationships by leveraging distinct geodesic distances.

6.2.4.1 Euclidean Relationship

The Euclidean space serves as a prominent example of a flat manifold, exhibiting zero curvature across all points. Within Euclidean space, the calculation of the geodesic distance between any two points is given by the conventional Euclidean distance formula. The distance d_{euc} is the straight-line distance between two points $\mathbf{x}, \mathbf{y}$ in a Cartesian coordinate system given by:

$$d_{\text{euc}}(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\| \,. \tag{6.6}$$

In our work, the Euclidean distance yields the Euclidean-based losses as:

$$\mathcal{L}^{\text{euc}}_{\text{tt}-\text{ss}} = \sum_{i,j \in [B]} \ell(\ d_{\text{euc}}(\mathbf{t}_i, \mathbf{t}_j),\ d_{\text{euc}}(\mathbf{s}_i, \mathbf{s}_j)\ ), \tag{6.7}$$

$$\mathcal{L}^{\text{euc}}_{\text{ts}-\text{ss}} = \sum_{i,j \in [B]} \ell(\ d_{\text{euc}}(\mathbf{t}_i, \mathbf{s}_j),\ d_{\text{euc}}(\mathbf{s}_i, \mathbf{s}_j)\ ). \tag{6.8}$$

6.2.4.2 Spherical Relationship

The second relationship we consider is the spherical relationship. In contrast to Euclidean space, the spherical manifold displays a distinct characteristic by possessing a constant positive curvature. The geodesic distance between two points is calculated based on the angular separation between the points and the radius of the sphere. Following previous works [123, 125], we adopt the cosine distance to explore the spherical-based relationship, which is given by

$$d_{\cos}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \, \|\mathbf{y}\|}. \tag{6.9}$$

Then we incorporate the cosine distance as the second consideration to explore positive-curvature relationships, and the losses are formulated as:

$$\mathcal{L}^{\cos}_{\mathrm{tt-ss}} = \sum_{i,j \in [B]} \ell(d_{\cos}(\mathbf{t}_i, \mathbf{t}_j), d_{\cos}(\mathbf{s}_i, \mathbf{s}_j)), \tag{6.10}$$

$$\mathcal{L}^{\cos}_{\mathrm{ts-ss}} = \sum_{i,j \in [B]} \ell(d_{\cos}(\mathbf{t}_i, \mathbf{s}_j), d_{\cos}(\mathbf{s}_i, \mathbf{s}_j)). \tag{6.11}$$

6.2.4.3 Hyperbolic Relationship

A comprehensive relationship evaluation would benefit more from various feature pattern exploration of KD agents, and it can thus contribute to more effective KD from the teacher to the cross-modal student. However, the above two measurements explore feature relationships in either zero-curvature or positive-curvature manifolds as in RKD [71]. There is a lack of consideration of relationships in negative curvature manifolds, which would result in insufficient KD. To this end, we introduce the third relationship based on the negative curvature manifold.

In our work, we embed both teacher outputs and student outputs in the Poincaré ball, and the hyperbolic losses are computed as:

$$\mathcal{L}^{\mathrm{hyp}}_{\mathrm{tt-ss}} = \sum_{i,j \in [B]} \ell\Big(\, d_{\mathrm{hyp}}\Big(\mathbf{t}^{\mathrm{hyp}}_i, \mathbf{t}^{\mathrm{hyp}}_j\Big), d_{\mathrm{hyp}}\Big(\mathbf{s}^{\mathrm{hyp}}_i, \mathbf{s}^{\mathrm{hyp}}_j\Big) \, \Big), \tag{6.12}$$

$$\mathcal{L}^{\mathrm{hyp}}_{\mathrm{ts-ss}} = \sum_{i,j \in [B]} \ell\Big(\, d_{\mathrm{hyp}}\Big(\mathbf{t}^{\mathrm{hyp}}_i, \mathbf{s}^{\mathrm{hyp}}_j\Big), d_{\mathrm{hyp}}\Big(\mathbf{s}^{\mathrm{hyp}}_i, \mathbf{s}^{\mathrm{hyp}}_j\Big) \, \Big), \tag{6.13}$$

where $\mathbf{t}_i^{\text{hyp}} = \exp_0^{\text{c}}(\mathbf{t}_{\text{i}})$ and $\mathbf{s}_i^{\text{hyp}} = \exp_0^{\text{c}}(\mathbf{s}_{\text{i}})$ are hyperbolic teacher and student embeddings, respectively.

6.2.5 Overall Loss Function

Finally, we combine the insights from multiple agents and multiple manifolds to construct our distillation pipeline. Specifically, we first formulate two distillation losses, including the self-agent distillation loss $\mathcal{L}_{\text{KD-S}}$ and the cross-agent distillation loss $\mathcal{L}_{\text{KD-C}}$ respectively as:

$$\mathcal{L}_{\text{KD-S}} = \mathcal{L}_{\text{tt-ss}}^{\text{euc}} + \mathcal{L}_{\text{tt-ss}}^{\text{cos}} + \mathcal{L}_{\text{tt-ss}}^{\text{hyp}}, \tag{6.14}$$

$$\mathcal{L}_{\text{KD-C}} = \mathcal{L}_{\text{ts-ss}}^{\text{euc}} + \mathcal{L}_{\text{ts-ss}}^{\text{cos}} + \mathcal{L}_{\text{ts-ss}}^{\text{hyp}}. \tag{6.15}$$

Subsequently, with weight hyperparameters $\lambda_{\text{S}}, \lambda_{\text{C}}$ and the triplet loss as the visual place recognition task loss $\mathcal{L}_{\text{task}}$, we propose three different overall losses. They are denoted as DistilVPR-S, DistilVPR-C, and DistilVPR-SC, respectively:

$$\mathcal{L}_{\text{DistilVPR-S}} = \mathcal{L}_{\text{task}} + \lambda_{\text{S}}\mathcal{L}_{\text{KD-S}}, \tag{6.16}$$

$$\mathcal{L}_{\text{DistilVPR-C}} = \mathcal{L}_{\text{task}} + \lambda_{\text{C}}\mathcal{L}_{\text{KD-C}}, \tag{6.17}$$

$$\mathcal{L}_{\text{DistilVPR-SC}} = \mathcal{L}_{\text{task}} + \lambda_{\text{S}}\mathcal{L}_{\text{KD-S}} + \lambda_{\text{C}}\mathcal{L}_{\text{KD-C}}. \tag{6.18}$$

6.3 Experiments

In this section, we conduct experiments to compare DistilVPR defined in (6.16) to (6.18) with other KD baselines. We also provide necessary ablation studies to verify the design efficacy.

6.3.1 Datasets and Implementation Details

6.3.1.1 Oxford RobotCar

The Oxford RobotCar dataset [101] is a large-scale autonomous driving dataset, which provides a rich collection of sensor data, including images and point clouds. It also encompasses various driving scenarios with different weather conditions, traffic patterns, and pedestrian interactions. We use the processed point clouds provided by PointNetVLAD[4] which is the standard benchmark data for point cloud and multi-modal (image + point cloud) place recognition. Since it is equipped with both images and point clouds, the Oxford RobotCar dataset would be a suitable platform to test the performance of multi-modal teachers and single-modal students.

6.3.1.2 Boreas

The Boreas dataset [114] is gathered by conducting multiple drives along a consistent route over one year, thereby capturing notable seasonal fluctuations. It comprises an extensive collection of over 350 km of driving data, featuring numerous sequences recorded under challenging weather conditions, including rain, heavy snow, and night. It also provides multi-modal sensor data such as images and point clouds, and thus can also serve as a benchmark for both multi-modal and single-modal models.

6.3.1.3 Implementation Details

We choose two SOTA multi-modal place recognition models as teachers, including MinkLoc++ [8] and AdaFusion [66]. We use their single-modal branches as students to test the effectiveness of cross-modal KD. We use the Adam optimizer to train both teachers and students. The learning rate is set as $1e-4$ and $1e-3$ for the image branch and the point cloud branch respectively. Both teacher models and student models are trained for 60 epochs with 128 batch size. All experiments are conducted on an A100 GPU. We follow previous works to use the same evaluation protocol, including Average Recall@1 (AR@1) and Average Recall@1% (AR@1%).

6.3.2 Main Results

6.3.2.1 Fusion-to-single Distillation

As shown in Table 6.1 and Table 6.2, our proposed three KD schemes can achieve considerably better performance compared with other counterparts in the Oxford and the Boreas datasets. In addition, our schemes can handle various fusion-to-single KD tasks, including fusion-to-2D and fusion-to-3D, which further underscores the efficacy and generalization ability. We have also noticed that relational KD schemes generally outperform the direct KD counterparts, which shows that the key to effective distillation for visual place recognition lies in the exploration of feature relationships rather than mere feature alignment.

Moreover, we have found that the 3D point cloud inputs can always contribute better visual place recognition performance compared with the 2D image inputs. This trend holds across both datasets, with the gap being particularly pronounced in the more challenging Boreas dataset. This observation reinforces the assertion that utilizing point cloud data is pivotal in achieving effective visual place recognition results.

Distillation Method	T: MinkLoc++ S: MinkLoc++2D		T: MinkLoc++ S: MinkLoc++3D		T: AdaFusion S: AdaFusion-2D		T: AdaFusion S: AdaFusion-3D	
	AR@1%	AR@1	AR@1%	AR@1	AR@1%	AR@1	AR@1%	AR@1
Teacher	99.4	97.2	99.4	97.2	99.0	96.6	99.0	96.6
Student w/o distil.	94.7	85.7	98.1	94.4	94.2	84.2	98.0	93.8
*KD [70]	95.2	84.6	98.1	94.3	95.2	84.6	97.8	93.7
*AFD [72]	95.2	84.7	98.1	94.2	94.5	82.9	97.8	93.2
*EPC-Net [126]	95.4	85.6	97.9	93.8	95.3	85.1	98.1	93.7
RKD [71]	96.5	88.5	**98.3**	94.5	96.2	87.3	<u>98.2</u>	<u>94.4</u>
CSD [124]	95.4	86.0	98.1	94.4	95.3	85.4	98.0	93.8
LSD-Net [74]	95.8	86.1	98.2	94.1	95.9	86.2	97.9	93.9
MKD [73]	95.0	85.4	98.1	93.9	95.1	84.5	97.8	93.7
(ours) DistilVPR-S	96.7	88.7	**98.3**	**95.2**	96.2	87.4	98.0	94.2
(ours) DistilVPR-C	**97.3**	**91.1**	98.1	94.4	<u>96.6</u>	<u>88.8</u>	98.0	93.7
(ours) DistilVPR-SC	<u>97.0</u>	<u>90.0</u>	**98.3**	<u>94.6</u>	**96.7**	**89.0**	**98.3**	**94.7**

TABLE 6.1: Fusion-to-single distillation comparison on the Oxford RobotCar dataset. "T:" and "S:" stand for the teacher model and the student model respectively. Direct distillation solutions are marked with "*", while relational solutions are without any mark. The best results are bold and underlined, while the second-best results are underlined only.

Distillation Method	T: MinkLoc++ S: MinkLoc++2D AR@1%	AR@1	T: MinkLoc++ S: MinkLoc++3D AR@1%	AR@1	T: AdaFusion S: AdaFusion-2D AR@1%	AR@1	T: AdaFusion S: AdaFusion-3D AR@1%	AR@1
Teacher	98.9	93.1	98.9	93.1	98.9	93.2	98.9	93.2
Student w/o distil.	75.2	60.0	98.5	91.0	74.5	59.6	98.9	91.5
*KD [70]	75.8	61.4	98.1	90.4	76.9	60.3	98.5	91.7
*AFD [72]	75.5	60.4	97.4	88.4	75.9	58.5	98.7	92.7
*EPC-Net [126]	75.3	60.8	98.0	89.8	75.4	60.5	98.9	92.4
RKD [71]	78.0	62.9	**99.1**	91.6	78.8	62.5	98.9	93.9
CSD [124]	76.3	61.4	98.2	90.9	77.0	61.2	99.1	92.8
LSD-Net [74]	74.5	59.3	98.7	92.0	76.7	60.4	98.5	92.2
MKD [73]	77.6	61.3	96.9	88.2	76.5	61.0	98.9	92.2
(ours) DistilVPR-S	77.3	63.4	98.5	**92.1**	**80.1**	64.1	**99.3**	**94.0**
(ours) DistilVPR-C	77.0	65.1	97.8	90.5	79.7	64.7	98.8	92.3
(ours) DistilVPR-SC	**79.3**	**67.2**	98.3	91.3	78.0	**65.5**	**99.3**	93.6

TABLE 6.2: Fusion-to-single distillation comparison on the Boreas dataset.

6.3.2.2 3D-to-2D and Big-to-small Distillation

We evaluate the cross-modal distillation performance by training teachers with pure 3D point cloud inputs and students with pure 2D images. As illustrated in Table 6.3, the distinct advantages of DistilVPR become more evident in this context. Notably, in the 3D-to-2D scenarios, DistilVPR-SC exhibits notably superior performance compared to other baselines. This result underscores the pronounced effectiveness of our methodology in addressing the intricate challenge of distillation across disparate modalities. We also assess the basic scenario of distillation from a larger model to a smaller one, as presented in Table 6.3. In this setting, our proposed approach continues to demonstrate effective distillation performance.

Distillation Method	T: MinkLoc++2D-Big S: MinkLoc++2D AR@1%	AR@1	T: MinkLoc++3D S: MinkLoc++2D AR@1%	AR@1	T: AdaFusion-2D-Big S: AdaFusion-2D AR@1%	AR@1	T: AdaFusion-3D S: AdaFusion-2D AR@1%	AR@1
Teacher	80.3	66.4	98.5	91.0	87.8	64.7	98.9	91.5
Student w/o distil.	75.2	60.0	75.2	60.0	74.5	59.6	74.5	59.6
*KD [70]	75.1	60.1	74.9	58.6	76.8	60.6	75.8	59.0
*AFD [72]	77.3	62.5	75.3	56.5	76.2	58.5	76.5	59.4
*EPC-Net [126]	74.8	59.2	73.5	58.6	**77.3**	60.4	75.2	60.5
RKD [71]	76.4	61.7	76.2	60.4	75.8	61.5	77.4	61.4
CSD [124]	77.3	60.3	76.2	60.3	76.1	60.2	76.8	61.0
LSD-Net [74]	77.6	61.9	75.1	57.0	74.8	60.3	74.7	59.1
MKD [73]	77.2	60.1	75.5	59.0	74.1	60.0	75.5	60.1
(ours) DistilVPR-S	**77.9**	62.0	76.4	61.3	76.8	62.2	77.1	62.6
(ours) DistilVPR-C	77.0	64.2	78.0	66.4	**77.3**	64.2	78.4	65.9
(ours) DistilVPR-SC	77.1	**65.4**	**81.1**	**68.2**	76.8	**64.7**	**79.0**	**66.5**

TABLE 6.3: Big-to-small and 3D-to-2D distillation comparison on the Boreas dataset.

6.3.3 Ablation Studies

6.3.3.1 Agent Relationships

We compare the performance of different relationships as in Table 6.4. The combination of using both self-agent and cross-agent relationships achieves optimal performance, which verifies the effectiveness of our multi-agent relationships.

Method	AR@1%	AR@1
w/o distil.	75.2	59.3
$\mathcal{L}_{\text{tt}-\text{ss}}$ in (6.3)	76.4	61.3
$\mathcal{L}_{\text{ts}-\text{ss}}$ in (6.4)	78.0	66.4
$\mathcal{L}_{\text{tt}-\text{ts}}$ in (6.5)	76.1	60.6
$\mathcal{L}_{\text{tt}-\text{ss}} + \mathcal{L}_{\text{ts}-\text{ss}}$	**81.1**	**68.2**

TABLE 6.4: Ablation study on the self-agent and cross-agent relationship computation.

6.3.3.2 Manifold Relationships

We proceed to examine the utilization of different relationship distances, as detailed in Table 6.5. Notably, the three fundamental distances yield comparable individual performances. Further using only two manifold distances with insufficient curvature exploration could not always bring improvements compared with using a single manifold. By contrast, through the fusion of sufficient relationship distances across multiple manifolds with consideration of all types of curvatures, a remarkable enhancement in distillation performance is observed. This substantiates the effectiveness of our approach in exploiting feature relationships within diverse curvature manifolds.

6.3.3.3 Different Teacher Modalities

In Table 6.6, we present a comparison of the distillation performance achieved with different teachers. Intriguingly, it is observed that the 3D-based model MinkLoc++3D can even outperform the fusion model MinkLoc++ in terms of distillation efficiency. This finding underscores the notion that a good task performer might not necessarily translate into a good teacher for distillation.

d_{euc}	d_{cos}	d_{hyp}	Ours-S	Ours-C	Ours-SC
✓			59.9	65.2	66.8
	✓		60.2	64.9	66.5
		✓	60.0	65.6	67.0
✓	✓		60.5	65.8	67.4
✓		✓	60.2	66.0	66.8
	✓	✓	60.1	66.1	66.9
✓	✓	✓	**61.3**	**66.4**	**68.2**

TABLE 6.5: AR@1 comparison on different distance functions and relationship agent combinations.

Teacher	T: AR@1	S: AR@1
MinkLoc++	**93.1**	67.2
MinkLoc++3D	91.3	**68.2**
MinkLoc++2D-big	66.4	65.4

TABLE 6.6: Distillation from different teachers. The student is MinkLoc++2D with DistilVPR-SC.

6.3.3.4 Visualization

A more detailed example is illustrated in Fig. 6.3 with visualized salience maps. Distillation facilitates the student in emphasizing scene-specific objects such as buildings, which showcases the effectiveness of teacher knowledge.

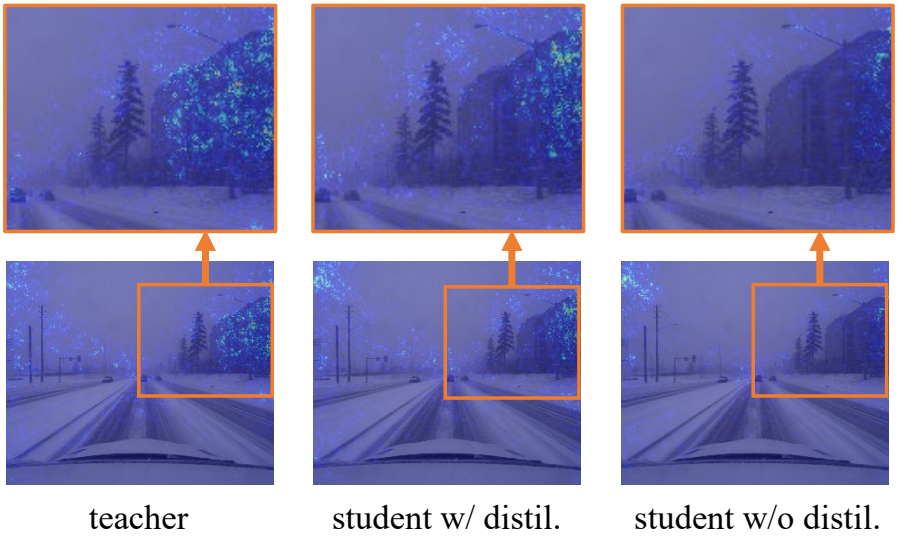

FIGURE 6.3: Visualization of the salience maps. With distillation from the teacher, the student is guided to focus on scene-specific objects such as buildings.

6.4 Conclusion and Limitations

This paper presents DistilVPR, a novel cross-modal distillation pipeline designed for enhancing visual place recognition. We harness multi-agent and multi-manifold relationships to facilitate knowledge exploration, leading to superior performance compared to other distillation baselines.

A limitation of our approach lies in its assumption of identical feature dimensions between teachers and students, potentially restricting its applicability. Nevertheless, this limitation could be addressed by employing a feature adaptor to align the feature dimensions of teachers and students.

Chapter 7

Cross-View Localization

In the previous Chapters 3 to 6, we mainly discuss and test ground-view localization scenarios that all reference data is from the ground systems. However, the collection of the ground reference data would be tedious due to limited ground FOV. By contrast, beyond same-view place recognition, utilizing aerial data such as satellite images and semantic maps introduces a cross-view approach to visual localization. Aerial data offers a significantly larger FOV and provides geo-referenced information from a top-down perspective, complementing the limited spatial context available from ground-level observations. In this aerial-ground place recognition framework, the ground-level data serves as the query, while the aerial data serves as the reference. A significant challenge arises from the domain gap between aerial and ground data, which is primarily due to their different views. Aligning cross-view features effectively remains a challenging issue in this problem.

The current aerial-ground multi-modal place recognition method [69] depends significantly on pre-computed semantic masks to construct object graphs based on detected objects. This reliance reduces flexibility in diverse or unstructured settings, especially when clear semantic information is either absent or noisy. Additionally, existing multi-modal models [8, 67] often overlook the importance of enhancing geometric awareness in the generated scene descriptors, a key factor for achieving accurate spatial reasoning in aerial-ground scenarios.

In light of these challenges, we propose a cross-view solution by constructing a surrogate manifold that bridges the feature domains, facilitating smoother feature

alignment across the aerial and ground modalities. Additionally, we leverage the non-intersection property of neural ODEs to generate geo-consistent scene descriptors. This approach ensures that the features we extract maintain consistency with actual geometric distances, which enhances cross-view place recognition and yields more accurate visual localization results.

7.1 Introduction

Apart from models addressing scenarios where both query data and database data come from the same viewpoint, achieving place recognition in cross-view scenarios (aerial-ground[127–130]) presents another interesting problem. The aerial-ground place recognition is particularly valuable for ground robot localization due to the availability of extensive aerial RGB or semantic maps that offer broad FOV coverage. Additionally, collecting aerial data is more efficient than ground data, especially in challenging terrains like forests and mountains. These aerial maps provide rich geometric references, enabling ground robots to achieve accurate and effective localization[131, 132].

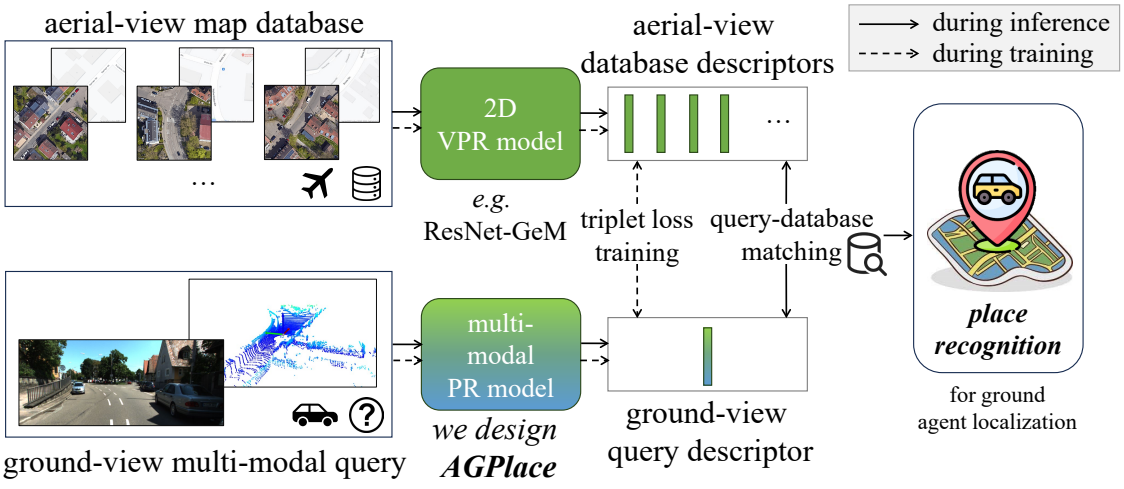

FIGURE 7.1: The pipeline of the multi-modal aerial-ground place recognition problem. (1) The aerial-view geo-tagged maps (e.g. aerial RGB images, road maps) act as the database; (2) The ground-view multi-modal data (images + point clouds) are the place query to be matched with the database.

We investigate the challenge of multi-modal place recognition in the aerial-ground scenario. More specifically, as shown in Fig. 7.1, we consider the ground data to be with multiple modalities, consisting of both images and point clouds. On the other

hand, aerial data encompasses various formats, including real captured aerial RGB images as well as semantic maps such as road maps.

Despite advancements, the existing multi-modal cross-view place recognition approach VSGP [69] relies heavily on pre-computed semantic predictions to build object graphs from detected objects. While this is a useful approach in structured environments, it limits adaptability in diverse and unstructured settings, particularly where clear semantic information is lacking or noisy. Furthermore, current multi-modal models [8, 67] often neglect to enhance the geometric awareness of the constructed scene descriptors, which hinders their performance in aerial-ground localization scenarios.

To mitigate these challenges, we utilize a manifold to fuse multi-modal information by learning manifold chart functions to relate points in 2D or 3D spaces to the manifold. The fusion embedding is constructed using neural ODEs, which describe the movement of points on the manifold. Subsequently, the updated fusion embedding acts as guidance for the extraction of respective modal features to formulate the final scene descriptor. We summarize our contributions as follows:

1. We propose AGPlace, a multi-modal solution to deal with aerial-ground place recognition. It leverages neural ODEs to achieve effective multi-sensor feature fusion inspired by point movement on a manifold.
2. We establish two benchmarks to evaluate AGPlace, which consists of different multi-sensor ground data and various aerial database maps.

7.2 Methodology

In this section, we first present the problem formulation of multi-modal aerial-ground place recognition, followed by detailed descriptions of our proposed AGPlace approach.

7.2.1 Modal-Specific Feature Extraction

The overall pipeline is shown in Fig. 7.2. We follow the previous cross-view VPR solution[128, 133] that uses two different networks (i.e., not sharing weights) to

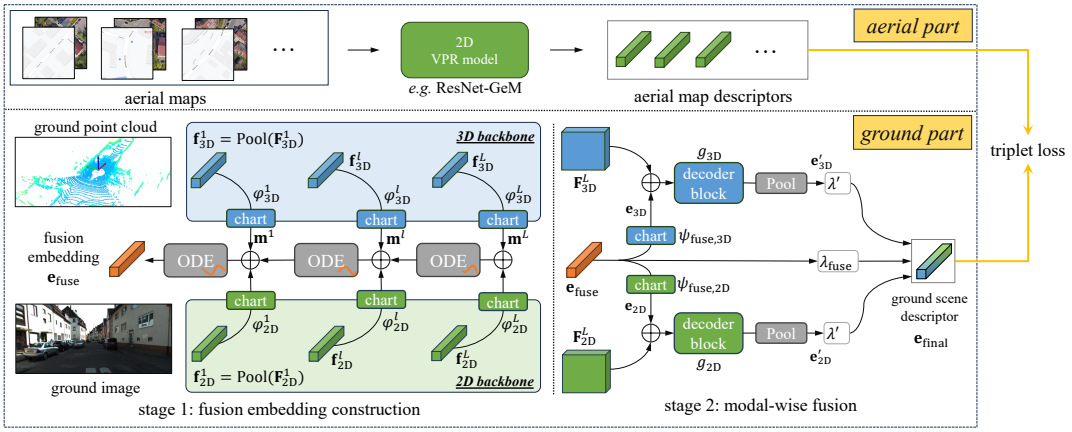

FIGURE 7.2: The pipeline overview. In the first stage, the ground image and point cloud are processed with separate backbone branches, the output features of which are used to build the fusion embedding. In the second stage, the constructed fusion embedding is mapped into the respective modal spaces to achieve further modal-wise feature extraction. The obtained ground and aerial descriptors are trained with the triplet loss.

handle aerial and ground inputs respectively. For the aerial part, since VPR for the 2D modality is well-established, we can use off-the-shelf 2D VPR methods (e.g., ResNet-GeM[43]) to extract aerial descriptors. And our design is mainly focusing on the ground part that takes multi-modal query inputs. Given 2D image and 3D point cloud query inputs, we first use separate 2D and 3D backbones to extract the respective basic features in each modality. Using hierarchical information is a common technique in various tasks, and we follow this manner to extract multiple feature maps. We denote the feature maps at the l-th backbone block as $\mathbf{F}^l_{\text{2D}}$ and $\mathbf{F}^l_{\text{3D}}$, and all the gathered feature maps are denoted respectively as:

$$\mathbf{F}_{\text{2D}} = \left\{ \mathbf{F}^l_{\text{2D}} \in \mathbb{R}^{N^l \times C} \ : \ l \in [L] \right\}, \tag{7.1}$$

$$\mathbf{F}_{\text{3D}} = \left\{ \mathbf{F}^l_{\text{3D}} \in \mathbb{R}^{N^l \times C} \ : \ l \in [L] \right\}, \tag{7.2}$$

where N^l is the number of feature vectors at block l, C is the number of corresponding feature channels,[1] and L is the number of backbone blocks.[2]

For images, we use the standard 2D format. For point clouds, we use the 3D voxels format, which is a common format in 3D place recognition[8]. 3D point clouds can also be projected onto 2D planes and extracted by 2D backbones, e.g. spherical

[1] For illustration simplicity, we assume 2D and 3D backbones have the same number of feature vectors and channels.

[2] We denote $[N] = \{1, \ldots, N\}$.

projection and BEV projection. However, these projection methods would either lose information or change the neighboring connections, which would result in insufficient feature extraction.

7.2.2 Stage 1: Fusion Embedding Construction

To achieve effective information fusion between different modalities in multi-modal place recognition, we employ a two-stage fusion strategy as depicted in Fig. 7.2. In this first stage, in addition to the vanilla 2D and 3D feature branches, we introduce a third branch as a surrogate manifold for fusing feature information from the 2D and 3D branches and constructing the fusion embedding.

On the other hand, in the problem of aerial-ground place recognition, the significant domain gap between aerial and ground data poses a challenge for domain alignment. Introducing a third branch can act as a platform to help bridge this gap. The visualization of descriptor distances, shown in Fig. 7.3, demonstrates that the inclusion of the fusion manifold aids in better aligning the two domains, resulting in the aerial-ground scene descriptor distance that more closely corresponds to the actual geometric distance.

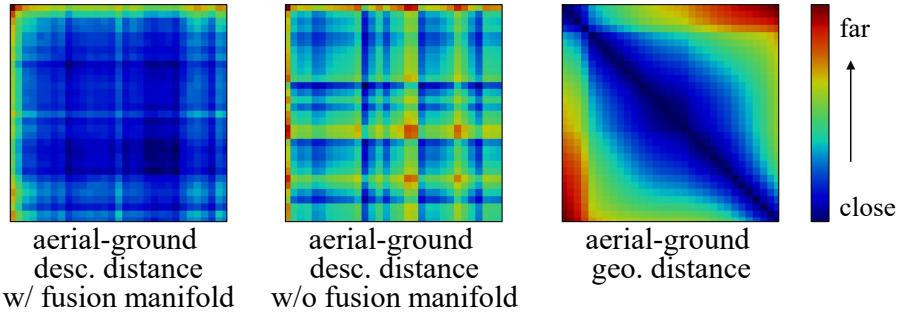

FIGURE 7.3: Aerial-ground (database-query) scene descriptor distance visualization. Symmetrization is applied for better visualization.

At the fusion embedding construction stage, we leverage the extracted feature maps $\mathbf{F}_{2D}$ and $\mathbf{F}_{3D}$ from each modal backbone to build the fusion embedding. Inspired by the concept of space point motion on manifolds in differential geometry [90], we model the fusion embedding construction as a multi-block state evolution process (shown in Fig. 7.4). This process starts from the last block and progresses towards the first block (i.e., $L \rightarrow 1$), allowing the embedding state to incorporate more global-level information at later blocks that is beneficial for the place recognition task. At

each block, the evolution consists of two sub-processes: (1) state initialization and (2) state updating.

7.2.2.1 State Initialization

At each backbone block l, we denote the initial fusion state as $\gamma^l(t=0) \in \mathcal{M}^C$ on the C-dimensional manifold $\mathcal{M}^C$, and define it as:

$$\gamma^l(0) = \begin{cases} \mathbf{m}^l & \text{if } l = L, \\ \mathbf{m}^l + \gamma^{l+1}(T) & \text{otherwise,} \end{cases} \tag{7.3}$$

where T is the state updating end time, and $\mathbf{m}^l \in \{\mathbf{m}^l\}_{l=1}^L$ is the fusion state momentum. Specifically, $\mathbf{m}^l$ is constructed using feature vectors from the respective l-th backbone block:

$$\mathbf{m}^l = \varphi_{2\mathrm{D}}^l(\mathbf{f}_{2\mathrm{D}}^l) + \varphi_{3\mathrm{D}}^l(\mathbf{f}_{3\mathrm{D}}^l). \tag{7.4}$$

Here, $\varphi_{2\mathrm{D}}^l(\cdot), \varphi_{3\mathrm{D}}^l(\cdot) : \mathbb{R}^C \to \mathcal{M}^C$ are learnable manifold chart functions representing connections between the 2D/3D space and the fusion manifold. A chart for a manifold $\mathcal{M}$ provides a local coordinate system, capturing local structures of $\mathcal{M}$. $\mathbf{f}_{2\mathrm{D}}^l, \mathbf{f}_{3\mathrm{D}}^l \in \mathbb{R}^C$ in (7.4) are the pooled 2D and 3D features from the respective modal space:

$$\mathbf{f}_{2\mathrm{D}}^l = \mathrm{Pooling}(\mathbf{F}_{2\mathrm{D}}^l), \quad \mathbf{f}_{3\mathrm{D}}^l = \mathrm{Pooling}(\mathbf{F}_{3\mathrm{D}}^l). \tag{7.5}$$

The pooling function can highly summarize the global information of the whole scene, which is beneficial for constructing a fusion embedding with rich global multi-modal domain representations and thus benefits the place recognition task. This differs from dense prediction tasks (e.g. depth estimation, segmentation) that more dig into local patterns.

7.2.2.2 State Updating

In Riemannian geometry, the movement of a point on a manifold can be described by a parametric curve [90]. Inspired by this, we model the process of updating the fusion embedding state as the movement of a point in the manifold.

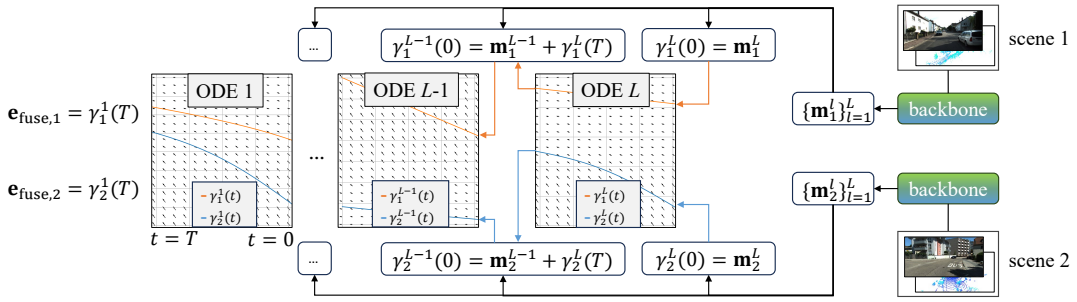

FIGURE 7.4: Fusion embedding evolution process starting from the last block and ending at the first block (i.e. $L \rightarrow 1$). The trajectories contain the movement of fusion embedding and are depicted by neural ODEs. In each ODE block, different inputs at time 0 guarantee different outputs at time T.

ODEs serve as a fundamental tool for describing dynamical systems. They express how a system state evolves over time, capturing relationships between the system variables and the rates of change. Based on this, the movement dynamics on a manifold can be described by an ODE:

$$\frac{d\gamma(t)}{dt} = f(\gamma(t), t), \tag{7.6}$$

where $\gamma(t)$ denotes the state trajectory.

The non-intersection of ODE solutions presented in Theorem 1 indicates that different ODE inputs guarantee different ODE outputs. This property helps to construct a more effective embedding that benefits place recognition. In our pipeline in (7.3), given that the initial fusion states at block l are different for different places, neural ODEs can update feature representations while theoretically maintain the feature differences at this block, which help constitute different scene descriptors and benefits the place recognition task (Fig. 7.4).

In detail, we update the fusion embedding from its initial state using parameterized neural ODEs at each block l, which can be regarded as point movements on the manifold:

$$\frac{d\gamma^l(t)}{dt} = f_{\theta^l}(\gamma^l(t)), \tag{7.7}$$

where $\gamma^l(t)$ is the fusion embedding state trajectory at block l, $f_{\theta^l}(\cdot)$ is a non-linear neural block with learnable parameters θ^l that describes the trajectory dynamics. By solving (7.7), we can obtain the updated state $\gamma^l(T)$ at the end time T. At the next block $l-1$, the start state $\gamma^{l-1}(0)$ is initialized based on momentum $\mathbf{m}^{l-1}$ and

the last end state $\gamma^l(T)$, as shown in Fig. 7.4 and (7.3). After progressive updating, finally, we obtain the end state at the first block $\gamma^1(T)$ as the final constructed fusion embedding $\mathbf{e}_{\text{fuse}} = \gamma^1(T) \in \mathcal{M}^C$.

Corollary 1. (Distinguished fusion states.) In each neural ODE block l, given two ODE inputs $\gamma_1^l(0) \neq \gamma_2^l(0)$ from two different scenes are different, then the two ODE outputs at this ODE block are different $\gamma_1^l(T) \neq \gamma_2^l(T)$.

As stated in Corollary 1, the neural ODE process can produce different fusion states, which would help construct more distinguished representations for different places and thus contribute to more effective place recognition performance. The visualized illustration of Corollary 1 is shown in Fig. 7.4.

7.2.3 Stage 2: Modal-Wise Fusion

The fusion embedding $\mathbf{e}_{\text{fuse}}$ captures rich multi-modal fusion information, which in turn serves as effective guidance for individual fine-grained local-level modal feature learning.

With the surrogate fusion embedding injected into the respective 2D/3D spaces, the tight coupling connection of the 2D and 3D embeddings could be softened, where the perturbation of a modality could be mitigated by the surrogate fusion information. As shown in Table 7.2, with the help of included fusion embedding in this stage 2, the final scene descriptor could be more robust against sensor failing.

Specifically, as illustrated in Fig. 7.2, we project the fusion embedding into the corresponding modal space to obtain 2D and 3D embeddings:

$$\mathbf{e}_{\text{2D}} = \psi_{\text{fuse,2D}}(\mathbf{e}_{\text{fuse}}), \qquad \mathbf{e}_{\text{3D}} = \psi_{\text{fuse,3D}}(\mathbf{e}_{\text{fuse}}), \tag{7.8}$$

where $\psi_{\text{fuse,2D}}(\cdot), \psi_{\text{fuse,3D}}(\cdot) : \mathcal{M}^C \to \mathbb{R}^C$ are chart functions that establish connections from the fusion manifold to the respective 2D and 3D spaces. These 2D and 3D embeddings capture rich multi-modal information, which is further combined with the original modal feature maps $\mathbf{F}_{\text{2D}}^L$ and $\mathbf{F}_{\text{3D}}^L$ from the last block L to construct

more effective representations through decoder blocks $g_{2D}(\cdot)$, $g_{3D}(\cdot)$:

$$\mathbf{e}'_{2D} = \text{Pooling}(g_{2D}(\mathbf{e}_{2D} \oplus \mathbf{F}^L_{2D})), \tag{7.9}$$

$$\mathbf{e}'_{3D} = \text{Pooling}(g_{3D}(\mathbf{e}_{3D} \oplus \mathbf{F}^L_{3D})), \tag{7.10}$$

where $\oplus$ denotes broadcast addition. By injecting the highly summarized fusion embedding, the respective local modal feature maps can capture more global information and achieve more robust representations. The decoded embeddings $\mathbf{e}'_{2D}$ and $\mathbf{e}'_{3D}$ are used to construct the final scene descriptor in the sequel.

7.2.4 Final Scene Descriptor and Loss Function

Our final ground scene descriptor consists of the decoded modal embeddings, as well as the fusion embedding. It is obtained as follows:

$$\mathbf{e}_{\text{final}} = \lambda'(\mathbf{e}'_{2D} + \mathbf{e}'_{3D}) + \lambda_{\text{fuse}}\mathbf{e}_{\text{fuse}}, \tag{7.11}$$

where $\lambda', \lambda_{\text{fuse}}$ are weights to balance different components.

As shown in Fig. 7.2, the domain connection between aerial and ground descriptors is achieved by minimizing the triplet loss function[112]:

$$\mathcal{L} = \max\big(\left\|\mathbf{e}^{\text{anchor}} - \mathbf{e}^{\text{positive}}\right\| - \left\|\mathbf{e}^{\text{anchor}} - \mathbf{e}^{\text{negative}}\right\| + m, 0\big), \tag{7.12}$$

where m is the margin hyperparameter and $\|\cdot\|$ is the ℓ_2-norm. $\mathbf{e}^{\text{anchor}}, \mathbf{e}^{\text{positive}}, \mathbf{e}^{\text{negative}}$ are the descriptors of an anchor query (ground), a positive map (aerial), and a negative map (aerial), respectively.

7.3 Experiments

7.3.1 Datasets and Implementation Details

To evaluate the performance of multi-modal aerial-ground place recognition models, we create benchmark datasets that consist of comprehensive ground sensor data and corresponding aerial database maps. We specifically construct two datasets, namely

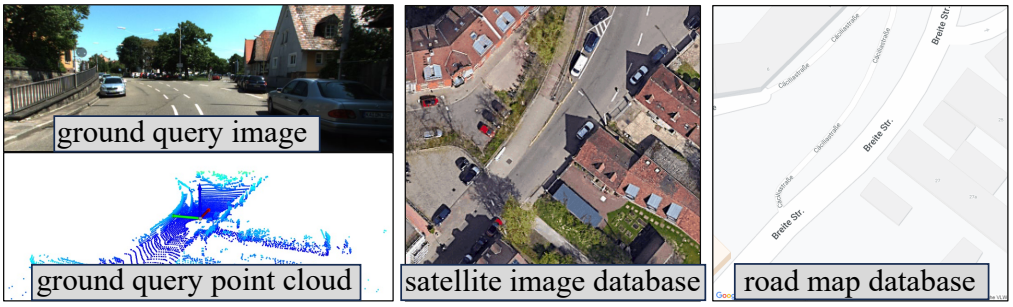

FIGURE 7.5: Data visualization from the KITTI360-AG dataset.

KITTI360-AG and *NuScenes-AG*. Examples are shown in Fig. 7.5. In addition, we also test our models on the Oxford RobotCar benchmark datasets used in previous multi-model ground-ground place recognition evaluations.

7.3.1.1 KITTI360-AG

For the ground dataset, we use the vanilla KITTI360 [134] dataset as the base to provide ground-view images, point clouds, and GNSS coordinates. We use a total of 7 sequences, in which the first 85% frames are for training and the last 15% frames are for testing, such that the test frames consist of seen and unseen areas for both fitting and generalization evaluation. The aerial data is generated based on the GNSS coordinates of the ground data frames. We use the Google Maps Static API[3] to download the corresponding aerial images, including satellite images, and road maps. We set the Google Maps Static API parameters as follows: scale=1, zoom=20, size=640×640. Each aerial image covers an approximately 75×75 m^2 area.

7.3.1.2 NuScenes-AG

The NuScenes [135] dataset provides a rich collection of sensor data collected from real urban driving scenarios, which also contains multi-view cameras. We download the aerial data based on GNSS coordinates with similar configurations mentioned in KITTI360-AG. The official train/test split is used in our setting, such that test scenes contain seen and unseen areas for both fitting and generalization evaluation.

[3] `https://developers.google.com/maps/documentation/maps-static/overview`

7.3.1.3 Oxford RobotCar

The Oxford RobotCar benchmark dataset is a public benchmark used for multi-modal ground-ground place recognition. There are two different splits used by MinkLoc++ (Oxford-Mink+) and AdaFusion (Oxford-Ada) respectively. And we compare our model under both settings.

7.3.1.4 Implementation Details

We select open-sourced SOTA multi-modal place recognition models as ground network baselines for query descriptor construction, mainly including MinkLoc++ [8], AdaFusion [66], HMPR [136], LCPR [67], UMF [18], and MSSPlace [137]. VSGP-place recognition [69] does not provide open-source codes and EINet [138] only provides data splits without model codes, which hinders us from comparing with them. Our backbones are constructed based on MinkLoc++. For the aerial database, all 3D and multi-modal baselines use ResNet-18-GeM for a fair comparison (with a linear layer to align descriptor dimensions) as the aerial model. Both aerial and ground models are trained during training. We use recall@K ($K = 1, 5, 10$)(R@K), average recall@1%(AR@%1), and average recall@1(AR@1) as the metrics. The positive retrieval threshold is set as 25 m. All experiments are conducted on a Tesla A100 GPU. More dataset, baseline, and implementation details are provided in the supplement.

7.3.2 Main Results

7.3.2.1 KITTI360-AG (Satellite Images and Road Maps)

We conduct an evaluation of AGPlace on the KITTI360-AG dataset as demonstrated in Table 7.1 (Best is in **bold** and second best is underline). AGPlace outperforms all other baselines in both satellite and road map settings. On the other hand, when using the road map aerial database, most models can already show considerable recall performance. This suggests that semantic maps, which are easily accessible and created without the need for satellites, drones, or airplanes, can already provide sufficient information for practical place recognition tasks. This finding opens up interesting possibilities for future research in the field [139, 140]. We also test the

2D VPR methods with the strong foundation backbone DINOV2[141], some of which can even surpass 3D counterparts.

Model	Satellite			Road Map		
	R@1	R@5	R@10	R@1	R@5	R@10
(2018) GeM (ResNet18)	22.1	34.8	44.5	20.5	31.9	37.5
(2022) TransGeo (DINOV2)	31.1	46.1	53.9	19.9	33.1	42.9
(2023) AnyLoc* (DINOV2)	4.2	8.7	13.0	5.4	5.5	5.9
(2024) SALAD (DINOV2)	31.4	46.3	53.2	20.1	32.4	39.1
(2022) MinkLoc3DV2	27.5	38.5	44.7	26.0	37.7	43.7
(2023) BEVPlace	21.7	34.8	42.1	19.4	33.8	41.1
(2021) MinkLoc++	31.9	46.2	52.0	28.8	43.0	50.3
(2022) AdaFusion	33.0	44.9	51.9	28.2	42.3	50.7
(2024) LCPR	33.3	48.1	54.0	26.0	42.9	51.1
(2024) UMF	31.1	46.6	54.4	29.1	43.5	51.4
(2024) MSSPlace	32.2	46.9	53.0	29.0	43.4	50.5
AGPlace (ours)	**35.7**	**50.5**	**57.0**	**30.5**	**45.6**	**53.9**

TABLE 7.1: Aerial-ground place recognition results on the KITTI-360-AG dataset using satellite or road map aerial sources. "*" denotes the model is frozen and purely relies on pre-trained weights.

7.3.2.2 NuScenes-AG (Ground Sensor Failing)

We next experiment on the NuScenes-AG dataset that consists of a LiDAR and multiple cameras, where AGPlace can also achieve significant performance. With the fusion embedding injected into the respective 2D/3D spaces in stage 2, the final scene descriptor demonstrates stronger robustness both camera and LiDAR failing. This demonstrates that our model can be more applicable in real deployment conditions.

Model	cams + LiDAR		cams fail		LiDAR fail	
	R@1	R@5	R@1	R@5	R@1	R@5
MinkLoc++	70.4	82.1	16.9	27.3	2.6	7.3
AdaFusion	71.9	82.3	9.0	17.8	3.5	6.7
LCPR	57.7	74.2	3.2	6.2	8.6	17.1
UMF	69.4	82.5	2.7	7.3	0.8	2.9
MSSPlace	71.3	82.2	17.4	29.8	5.0	8.7
AGPlace w/o stg. 2	72.8	83.3	18.8	29.6	6.4	15.7
AGPlace (ours)	**74.4**	**84.7**	**22.5**	**32.4**	**12.5**	**22.1**

TABLE 7.2: Aerial-ground place recognition results on the NuScenes-AG dataset using satellite image database. "fail" denotes dropping the modality input during testing. All models are trained with both modalities.

7.3.2.3 Oxford Benchmark

We also compare our model with previous SOTA baselines on the public Oxford benchmark dataset for ground-ground place recognition. As in Table 7.3, our model (without using extra data, re-ranking, or multiple cameras) can also show better performance than previous SOTA counterparts, which further verifies the effectiveness of our design.

Model	Extra train data	Oxford-Mink+		Oxford-Ada	
		AR@1%	AR@1	AR@1%	AR@1
CORAL		96.1	-	-	-
PIC-Net		98.2	-	-	-
Cues-Net		-	-	-	98.0
MinkLoc++		99.1	96.7	-	-
AdaFusion		-	-	99.2	98.2
HMPR		-	-	99.6	96.9
UMF	✓	99.1	97.9	-	-
MSSPlace		99.1	97.6	-	-
HMPR + *re-rank*		-	-	<u>99.7</u>	<u>99.0</u>
UMF + *re-rank*	✓	99.3	**98.3**	-	-
MSSPlace + *mul-cam*		<u>99.5</u>	98.2	-	-
AGPlace (ours)		**99.7**	**98.3**	**99.9**	**99.3**

TABLE 7.3: Ground-ground place recognition Results on the Oxford benchmark datasets. "-" denotes the result is not provided in the paper.

7.3.3 Ablation Studies

7.3.3.1 Module Ablation

We verify the effectiveness by ablating each module in our network. As shown in Table 7.4, both the two fusion stages in Fig. 7.2 can contribute to better final performance, confirming the efficacy of our approach.

7.3.3.2 ODE Updating

During the state evolution process, we parameterize the fusion embedding state trajectories as learnable neural ODEs. Notably in Table 7.5, upon removing the ODE updating mechanism, performance deteriorates across all metrics. ODEs also perform better than MLP and attention, both of which cannot guarantee outputs'

Stage 1	Stage 2	KITTI360-AG (Sate.) R@1	R@5	R@10
		31.9	46.2	52.0
✓		34.0	48.6	55.9
	✓	32.5	47.4	54.2
✓	✓	**35.7**	**50.5**	**57.0**

TABLE 7.4: Module ablation.

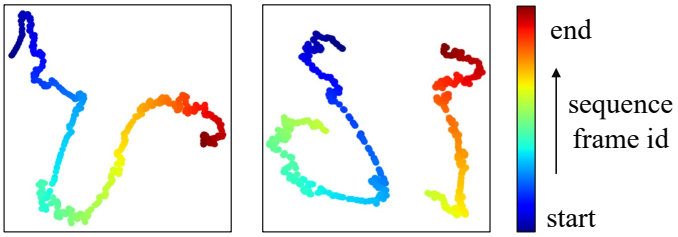

FIGURE 7.6: t-SNE plots of ground scene descriptors (from a consecutive frame sequence). ODEs can help build more consistent descriptors that align with the consecutive geometry.

differences. In addition, the t-SNE visualization in Fig. 7.6 shows ODEs can build more consecutive scene descriptors, underscoring the necessity of the ODE-based fusion state updating process for place recognition.

Next, we evaluate the performance of different fusion directions, as outlined in Table 7.5. The direction "block $L \rightarrow$ block 1", which transfers information from high-level summaries to detailed low-level representations, emerges as the preferred choice for fusion embedding construction.

State Updating Method	KITTI360-AG (Sate.) R@1	R@5	R@10
w/o ODE	33.5	48.2	55.3
w/ MLP	33.9	48.8	56.1
w/ attention	34.2	49.1	55.9
w/ ODE	**35.7**	**50.5**	**57.0**
Fuse Direction			
block 1 $\rightarrow$ block L	34.6	49.4	56.1
block 1 $\leftarrow$ block L	**35.7**	**50.5**	**57.0**

TABLE 7.5: State updating method and fusion direction.

Ground Modality	Aerial Modality	KITTI360-AG R@1	R@5	R@10
image	satellite	28.5	47.0	54.2
point cloud	satellite	27.4	41.9	49.6
image + point cloud	satellite	**35.7**	**50.5**	**57.0**
image + point cloud	roadmap	29.2	45.5	53.9
image + point cloud	satellite + roadmap	**37.0**	**52.3**	**59.8**

TABLE 7.6: Ground and aerial modality comparison.

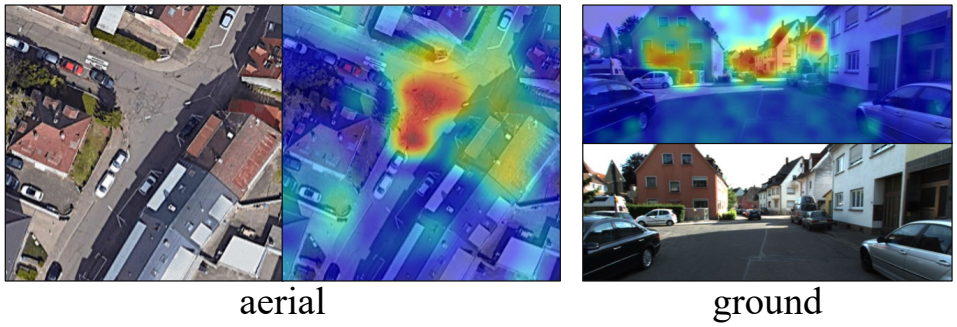

FIGURE 7.7: Salience map visualization.

7.3.3.3 Ground and Aerial Modalities

We evaluate the performance of various modality inputs, including pure 2D, pure 3D, and combined 2D+3D inputs, as illustrated in Table 7.6. Our findings demonstrate that leveraging multiple sensors leads to significant performance improvements compared to single-sensor inputs, highlighting the effectiveness of multi-modal inputs. Moreover, we also explore fusing satellite images and road maps for more representative aerial descriptors. As in Table 7.6, the fusion on both ground and aerial sides gives even better performance, which indicates an interesting research topic for aerial fusion in the future. To investigate which part of ground and aerial inputs are focused in the place recognition network, we visualize the salience maps as shown in Fig. 7.7. The geometric landmarks are emphasized (e.g. buildings and roads), which verifies the geo-feature extraction effectiveness of both aerial and ground models.

7.3.3.4 Runtime Performance

Finally, we assess the runtime speed and memory consumption of our model. AG-Place demonstrates acceptable real-time performance, satisfying basic deployment

and application requirements. All compared models exhibit comparable memory usage, with no significant discrepancies observed in this metric.

Model	Speed (FPS)	GPU Memory Usage (GB)
MinkLoc++	80	**0.58**
AdaFusion	72	0.59
LCPR	**125**	0.72
UMF	70	0.60
MSSPlace	75	0.59
AGPlace (ours)	62	0.61

TABLE 7.7: Runtime performance comparison on a Tesla A100.

7.4 Conclusion and Limitations

In this work, we have proposed AGPlace, a multi-modal model for addressing the aerial-ground place recognition problem. AGPlace leverages neural ODEs rooted in differential geometry to achieve effective multi-modal feature interaction. Our experiments demonstrate that AGPlace outperforms previous baselines and performs well in large-scale scenes, making it suitable for deployment in real-world environments.

However, this work primarily focuses on ground query data collected in urban environments, which may limit its generalizability to other realistic settings. In future work, we plan to expand our dataset and model to include more diverse and challenging scenarios, such as forests, deserts, and coastlines. This will enable us to establish a more comprehensive and generalized multi-modal cross-view place recognition pipeline.

Chapter 8

Conclusion and Future Work

In this chapter, we provide a summary of the visual localization solutions introduced throughout this thesis. We revisit the key methodologies and contributions. Additionally, we offer a discussion on potential future research directions, exploring areas where further development could lead to significant improvements in visual localization techniques. This includes identifying emerging challenges and opportunities for innovation, which may help address current limitations and expand the applicability of these solutions to new and diverse contexts.

8.1 Conclusion

In this thesis, we investigate the solutions for robust and effective visual localization. First, we propose to leverage different sensors, including cameras and LiDARs, to achieve both single-modal and multi-modal localization. Second, we dig into feature representations and propose to leverage neural ODEs and manifolds to enhance the robustness of feature extraction, which contributes to more robust localization. Then, we propose relational KD such that the effectiveness of localization pipelines can be significantly boosted by stronger teachers. Finally, we also investigate the problem of cross-view localization by leveraging aerial maps to guide ground system localization effectively. The detailed contribution of each chapter is listed as follows.

In Chapter 3, we introduce RobustLoc, a camera pose regression model specifically designed for robustness in challenging driving conditions. RobustLoc enhances performance by utilizing covisible image information and neural graph diffusion, which enables the model to gather and integrate information from nearby sources. This strategy improves its ability to operate effectively in complex driving environments. Comprehensive experiments confirm that RobustLoc delivers SOTA results.

In Chapter 4, we present HypLiLoc, a network designed for LiDAR-based pose regression. The model excels at feature extraction by utilizing global graph attention, hyperbolic-Euclidean interaction, and modality-specific learning. HypLiLoc achieves SOTA performance on both outdoor and indoor datasets. While its effectiveness has been validated on city-scale datasets, further testing is required in more challenging scenarios, such as noise perturbations and adversarial attacks, to ensure its suitability for practical applications.

In Chapter 5, we introduce two multi-modal place recognition models, PRFusion and PRFusion++, that utilize manifold-based attention to enhance feature fusion at both global and local levels. These models also incorporate neural Beltrami diffusion to improve the robustness of feature learning. Extensive experiments on large-scale benchmarks demonstrate the effectiveness of these designs. However, due to the separate feature extraction for 2D and 3D inputs, these models are more resource-intensive than several existing alternatives. Future research could focus on streamlining the pipeline to enable faster and more efficient place recognition.

In Chapter 6, we introduce DistilVPR, a cross-modal distillation framework aimed at improving visual place recognition. The model leverages multi-agent and multi-manifold relationships to enhance knowledge transfer, resulting in superior performance over existing distillation baselines. One limitation of this approach is its reliance on matching feature dimensions between teacher and student networks, which may limit its broader application. However, this challenge could be addressed by incorporating a feature adaptor, allowing for the alignment of feature dimensions between the teacher and student models.

In Chapter 7, we introduce AGPlace to achieve effective arial-ground cross-view place recognition. AGPlace leverages the non-interaction property of ODEs, which facilitates the construction of more consistent scene descriptors that better align with the actual geographic distance. We evaluate AGPlace on large-scale datasets,

demonstrating its superior performance compared to other multi-modal baselines across a variety of challenging scenarios.

Collectively, these contributions advance visual localization across multiple fronts, offering robust and effective solutions for a wide range of real-world applications.

8.2 Future Works

While this thesis presents advancements in robust and effective visual localization, there remain several open challenges and promising avenues for future research. Addressing these challenges will not only extend the applicability of current techniques but also pave the way for more generalized, scalable, and adaptable localization systems. Below, we outline two key areas where future work could make meaningful contributions.

8.2.1 Generalizable Visual Localization

One possible avenue is the exploration of self-supervised or unsupervised learning techniques to reduce the heavy reliance on labeled data and enhance generalization to previously unseen environments. Traditional visual localization methods often depend on large-scale, manually annotated datasets, which are costly to obtain and difficult to scale across diverse geographies and environmental conditions. In contrast, self-supervised approaches can leverage inherent structures in data, such as temporal continuity in videos, geometric consistency in multi-view imagery, or cross-modal correspondences between images and LiDAR scans, to learn useful representations without explicit supervision.

Such techniques hold great promise for learning robust and transferable scene representations that remain stable under appearance changes due to weather, lighting, or seasonal variations. In particular, dynamic or densely cluttered urban environments pose significant challenges for supervised methods due to frequent structural changes and occlusions. By enabling models to learn directly from raw sensory data, self-supervised and unsupervised approaches may offer greater adaptability and resilience, paving the way toward scalable, globally deployable

visual localization systems that require minimal human intervention during data collection and model training.

8.2.2 Scalability to Large-Scale and Global Localization

Scaling visual localization methods to operate reliably in large-scale and geographically diverse environments, such as city-wide navigation, inter-city mapping, or cross-country autonomous driving, remains a fundamental challenge. As the size and complexity of the environment grow, issues such as memory consumption, retrieval latency, and sensitivity to environmental variations (e.g., lighting, weather, seasonal changes) become increasingly prominent. Existing approaches often struggle to maintain both efficiency and accuracy under such a scale.

Future research could investigate hierarchical or multi-level localization strategies that decompose the global localization problem into more manageable sub-tasks. For example, a coarse-to-fine pipeline may first perform coarse region-level localization using global descriptors or GPS priors, followed by fine-grained pose estimation within the localized region. Additionally, leveraging scene graph hierarchies, semantic maps, or geo-spatial priors could further improve scalability and retrieval performance. These techniques, when combined with memory-efficient data structures and distributed indexing, may unlock new levels of scalability for real-world deployment.

8.2.3 Reasoning-Based Localization

Humans often localize themselves by reasoning over geo-relevant cues, such as landmarks, spatial relationships, and contextual knowledge, rather than solely relying on direct similarity comparisons between the current view and a database. This cognitive approach fundamentally differs from conventional similarity-based localization methods. Recently, large language models (LLMs) have exhibited strong reasoning capabilities in domains such as mathematical problem solving and code generation. Inspired by these developments, we can also explore the integration of LLMs into visual localization tasks to enable geo-reasoning based on visual cues. This integration allows the localization pipeline to go beyond appearance matching, offering improved interpretability, robustness, and effectiveness. Moreover, reasoning

enables the model to extract richer contextual information from the query image, such as the spatial relationships between multiple landmarks.

List of Publications, Patents and Codes

Publications

1. Quanjiang Guo, Jinchuan Zhang, **Sijie Wang**, Ling Tian, Zhao Kang, Bin Yan, Weidong Xiao. "Bridging generative and discriminative learning: Few-shot relation extraction via two-stage knowledge-guided pre-training." In Proceedings of the International Joint Conference on Artificial Intelligence, 2025.

2. **Sijie Wang**, Rui She, Qiyu Kang, Siqi Li, Disheng Li, Tianyu Geng, Shangshu Yu, and Wee Peng Tay. "Multi-modal aerial-ground cross-view place recognition with neural ODEs." In Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference, 2025.

3. Shangshu Yu, Xiaotian Sun, Wen Li, Qingshan Xu, Zhimin Yuan, **Sijie Wang**, Rui She, Cheng Wang. "STGC-NeRF: Spatial-temporal geometric consistency for LiDAR neural radiance fields in dynamic scenes." In Proceedings of the AAAI Conference on Artificial Intelligence, 2025.

4. Quanjiang Guo, Yihong Dong, Ling Tian, Zhao Kang, Yu Zhang, **Sijie Wang**. "BANER: Boundary-aware LLMs for few-shot named entity recognition." In Proceedings of the International Conference on Computational Linguistics, 2025.

5. **Sijie Wang**, Qiyu Kang, Rui She, Kai Zhao, Yang Song, and Wee Peng Tay. "PRFusion: Toward effective and robust multi-modal place recognition with image and point cloud fusion." IEEE Transactions on Intelligent Transportation Systems, 2024.

6. Rui She, **Sijie Wang**, Qiyu Kang, Kai Zhao, Yang Song, Wee Peng Tay, Tianyu Geng, and Xingchao Jian. "PosDiffNet: Positional neural diffusion

for point cloud registration in a large field of view with perturbations." In Proceedings of the AAAI Conference on Artificial Intelligence, 2024.

7. Qiyu Kang, Kai Zhao, Yang Song, Yihang Xie, Yanan Zhao, **Sijie Wang**, Rui She, and Wee Peng Tay. "Coupling graph neural networks with fractional order continuous dynamics: A robustness study." In Proceedings of the AAAI Conference on Artificial Intelligence, 2024.

8. **Sijie Wang**, Rui She, Qiyu Kang, Xingchao Jian, Kai Zhao, Yang Song, and Wee Peng Tay. "DistilVPR: Cross-modal knowledge distillation for visual place recognition." In Proceedings of the AAAI Conference on Artificial Intelligence, 2024.

9. Qiyu Kang, Wee Peng Tay, Rui She, **Sijie Wang**, Xiaoqian Liu, and Yuan-Rui Yang. "Multi-armed linear bandits with latent biases." Information Sciences, 2024.

10. Kai Zhao, Qiyu Kang, Yang Song, Rui She, **Sijie Wang**, and Wee Peng Tay. "Adversarial robustness in graph neural networks: A Hamiltonian approach." Advances in Neural Information Processing Systems, 2024.

11. Rui She, Qiyu Kang, **Sijie Wang**, Wee Peng Tay, Kai Zhao, Yang Song, Tianyu Geng, Yi Xu, Diego Navarro Navarro, and Andreas Hartmannsgruber. "PointDifformer: Robust point cloud registration with neural diffusion and transformer." IEEE Transactions on Geoscience and Remote Sensing, 2024.

12. Rui She, Qiyu Kang, **Sijie Wang**, Kai Zhao, Yang Song, Yi Xu, Tianyu Geng, Wee Peng Tay, Diego Navarro Navarro, and Andreas Hartmannsgruber. "Robust graph neural diffusion for image matching." In Proceedings of the IEEE International Conference on Image Processing, 2023.

13. Rui She, Qiyu Kang, **Sijie Wang**, Yuán-Ruì Yáng, Kai Zhao, Yang Song, and Wee Peng Tay. "Robustmat: Neural diffusion for street landmark patch matching under challenging environments." IEEE Transactions on Image Processing, 2023.

14. Rui She, Qiyu Kang, **Sijie Wang**, Wee Peng Tay, Yong Liang Guan, Diego Navarro Navarro, and Andreas Hartmannsgruber. "Image patch-matching with graph-based learning in street scenes." IEEE Transactions on Image Processing, 2023.

15. Qiyu Kang, Kai Zhao, Yang Song, **Sijie Wang**, and Wee Peng Tay. "Node embedding from neural Hamiltonian orbits in graph neural networks." In Proceedings of the International Conference on Machine Learning, 2023.

16. Kai Zhao, Qiyu Kang, Yang Song, Rui She, **Sijie Wang**, and Wee Peng Tay. "Graph neural convection-diffusion with heterophily." arXiv preprint arXiv:2305.16780 (2023).

17. **Sijie Wang**, Qiyu Kang, Rui She, Wei Wang, Kai Zhao, Yang Song, and Wee Peng Tay. "HypLiLoc: Towards effective LiDAR pose regression with hyperbolic fusion." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023.

18. Qiyu Kang, Kai Zhao, Yang Song, **Sijie Wang**, Rui She, and Wee Peng Tay. "Node embedding from Hamiltonian information propagation in graph neural networks." arXiv preprint arXiv:2303.01030 (2023).

19. Qiyu Kang, Yanan Kai Zhao Zhao, Xuhao Li, Qinxu Ding, Wee Peng Tay, and **Sijie Wang**. "Advancing graph neural networks through joint time-space dynamics." In The Symbiosis of Deep Learning and Differential Equations III, 2023.

20. **Sijie Wang**, Qiyu Kang, Rui She, Wee Peng Tay, Andreas Hartmannsgruber, and Diego Navarro Navarro. "RobustLoc: Robust camera pose regression in challenging driving environments." In Proceedings of the AAAI Conference on Artificial Intelligence, 2023.

21. Qiyu Kang, Rui She, **Sijie Wang**, Wee Peng Tay, Diego Navarro Navarro, and Andreas Hartmannsgruber. "Location learning for AVs: LiDAR and image landmarks fusion localization with graph neural networks." In Proceedings of the IEEE International Conference on Intelligent Transportation Systems, 2022.

22. **Sijie Wang**, Qiyu Kang, Rui She, Wee Peng Tay, Diego Navarro Navarro, and Andreas Hartmannsgruber. "Building facade parsing R-CNN." arXiv preprint arXiv:2205.05912 (2022).

23. Yang Song, Qiyu Kang, **Sijie Wang**, Kai Zhao, and Wee Peng Tay. "On the robustness of graph neural diffusion to topology perturbations." Advances in Neural Information Processing Systems, 2022.

24. Qiyu Kang, Kai Zhao, Yang Song, Yihang Xie, Yanan Zhao, **Sijie Wang**, Rui She, and Wee Peng Tay. "Coupling graph neural networks with non-integer order dynamics: A robustness study." In NeurIPS 2023 Workshop: New Frontiers in Graph Learning.

Under Review

1. **Sijie Wang** and Wee Peng Tay. "UAVScenes: A multi-modal dataset for UAVs." Anonymous Submission. (under review)

Patents

1. **Sijie Wang**, Rui She, Qiyu Kang, Tianyu Geng, and Wee Peng Tay, "AGPlace: Multi-modal place recognition in the aerial-ground cross view with differential geometry," Singapore Provisional Patent 10 202 401 872S, Jun. 25, 2024.

2. Rui She, Qiyu Kang, **Sijie Wang**, Wee Peng Tay, Navarro Diego Navarro, and Andreas Hartmannsgruber, "Computer-implemented method for estimating a position and/or pose of a vehicle," Germany Patent Application PCT/EP2024/061 019, Apr. 23, 2024.

3. **Sijie Wang**, Qiyu Kang, Rui She, Wee Peng Tay, Navarro Diego Navarro, and Andreas Hartmannsgruber, "A computer-implemented method for camera pose regression in a challenging traffic environment," UK Patent Application 2 216 510.4, Nov. 7, 2022.

Source Codes

1. RobustLoc `https://github.com/sijieaaa/RobustLoc`

2. HypLiLoc `https://github.com/sijieaaa/HypLiLoc`

3. PRFusion `https://github.com/sijieaaa/PRFusion`

4. DistilVPR `https://github.com/sijieaaa/DistilVPR`

5. AGPlace `https://github.com/sijieaaa/AGPlace`

Bibliography

[1] Shuanggen Jin, Estel Cardellach, and Feiqin Xie. *GNSS remote sensing*. Springer, 2014. 1

[2] Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: A convolutional network for real-time 6-DoF camera relocalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2938–2946, 2015. 1, 2, 3, 14, 29, 30

[3] Wei Wang, Bing Wang, Peijun Zhao, Changhao Chen, Ronald Clark, Bo Yang, Andrew Markham, and Niki Trigoni. PointLoc: Deep pose regressor for LiDAR point cloud localization. *IEEE Sensors Journal*, 22(1):959–968, 2021. 1, 4, 16, 47, 48, 51, 55, 56, 57, 58, 61

[4] Mikaela Angelina Uy and Gim Hee Lee. PointNetVLAD: Deep point cloud based retrieval for large-scale place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4470–4479, 2018. 2, 3, 4, 18, 57, 61, 73, 75, 96

[5] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12716–12725, 2019. 2

[6] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-perfect structure-from-motion with featuremetric refinement. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5987–5997, 2021. 2

[7] Paul-Edouard Sarlin, Ajaykumar Unagar, Mans Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, et al. Back to the feature: Learning robust camera localization from pixels to pose. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3247–3257, 2021. 2

[8] Jacek Komorowski, Monika Wysoczańska, and Tomasz Trzcinski. MinkLoc++: LiDAR and monocular image fusion for place recognition. In *Proceedings of the International Joint Conference on Neural Networks*, pages 1–8, 2021. 4, 20, 21, 63, 64, 66, 73, 75, 76, 77, 96, 103, 105, 106, 113

[9] Dragomir Anguelov, Carole Dulong, Daniel Filip, Christian Frueh, Stéphane Lafon, Richard Lyon, Abhijit Ogale, Luc Vincent, and Josh Weaver. Google street view: Capturing the world at street level. *Computer*, 43(6):32–38, 2010. 4

[10] Gabriele Berton, Alex Stoken, Barbara Caputo, and Carlo Masone. EarthLoc: Astronaut photography localization by indexing earth from space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12754–12764, 2024. 4

[11] Zhedong Zheng, Yunchao Wei, and Yi Yang. University-1652: A multi-view multi-source benchmark for drone-based geo-localization. In *Proceedings of the 28th ACM international conference on Multimedia*, pages 1395–1403, 2020. 4

[12] Panwang Xia, Yi Wan, Zhi Zheng, Yongjun Zhang, and Jiwei Deng. Enhancing cross-view geo-localization with domain alignment and scene consistency. *IEEE Transactions on Circuits and Systems for Video Technology*, 2024. 4

[13] Fabian Deuser, Konrad Habel, and Norbert Oswald. Sample4geo: Hard negative sampling for cross-view geo-localisation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16847–16856, 2023. 4

[14] Wansen Wu, Tao Chang, Xinmeng Li, Quanjun Yin, and Yue Hu. Vision-language navigation: a survey and taxonomy. *Neural Computing and Applications*, 36(7):3291–3316, 2024. 5

[15] Yiming Zhong, Qi Jiang, Jingyi Yu, and Yuexin Ma. Dexgrasp anything: Towards universal robotic dexterous grasping with physics awareness. *arXiv preprint arXiv:2503.08257*, 2025. 5

[16] Yanqing Shen, Turcan Tuna, Marco Hutter, Cesar Cadena, and Nanning Zheng. ForestLPR: LiDAR place recognition in forests attentioning multiple BEV density Images. *arXiv preprint arXiv:2503.04475*, 2025. 5

[17] Christopher Stewart, Michele Lazzarini, Adrian Luna, and Sergio Albani. Deep learning with open data for desert road mapping. *Remote Sensing*, 12 (14):2274, 2020. 5

[18] Alberto García-Hernández, Riccardo Giubilato, Klaus H Strobl, Javier Civera, and Rudolph Triebel. Unifying local and global multimodal features for place recognition in aliased and low-texture environments. *arXiv preprint arXiv:2403.13395*, 2024. 5, 20, 21, 75, 76, 77, 113

[19] Sijie Wang, Qiyu Kang, Rui She, Wee Peng Tay, Andreas Hartmannsgruber, and Diego Navarro Navarro. RobustLoc: Robust camera pose regression in challenging driving environments. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 6209–6216, 2023. 6, 57

[20] Sijie Wang, Qiyu Kang, Rui She, Wei Wang, Kai Zhao, Yang Song, and Wee Peng Tay. HypLiLoc: Towards effective LiDAR pose regression with hyperbolic fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5176–5185, 2023. 7, 18

[21] Sijie Wang, Rui She, Qiyu Kang, Xingchao Jian, Kai Zhao, Yang Song, and Wee Peng Tay. DistilVPR: Cross-modal knowledge distillation for visual place recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 10377–10385, 2024. 8

[22] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5974–5983, 2017. 14, 29, 41, 42

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 14, 51, 56

[24] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022. 14

[25] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 14, 26, 68

[26] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical vision Transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 14, 71

[27] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 14

[28] Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. Geometry-aware learning of maps for camera localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2616–2625, 2018. 14, 30, 41, 42, 57

[29] Zhaoyang Huang, Yan Xu, Jianping Shi, Xiaowei Zhou, Hujun Bao, and Guofeng Zhang. Prior guided dropout for robust visual localization in dynamic environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2791–2800, 2019. 14, 41, 57

[30] Bing Wang, Changhao Chen, Chris Xiaoxuan Lu, Peijun Zhao, Niki Trigoni, and Andrew Markham. AtLoc: Attention guided camera localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10393–10401, 2020. 15, 29, 41, 42, 57

[31] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. CoordiNet: uncertainty-aware pose regressor for reliable vehicle localization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2229–2238, 2022. 15, 29, 41, 42

[32] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. *Advances in Neural Information Processing Systems*, 31, 2018. 15

[33] Yuanming Hu, Baoyuan Wang, and Stephen Lin. Fc4: Fully convolutional color constancy with confidence-weighted pooling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4085–4094, 2017. 15

[34] Xu Yan, Jiantao Gao, Chaoda Zheng, Chao Zheng, Ruimao Zhang, Shuguang Cui, and Zhen Li. 2DPass: 2D priors assisted semantic segmentation on LiDAR point clouds. In *Proceedings of the European Conference on Computer Vision*, 2022. 15, 22

[35] Song Wang, Wentong Li, Wenyu Liu, Xiaolu Liu, and Jianke Zhu. LiDAR2Map: In defense of liDAR-Based semantic map construction using online camera distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5186–5195, 2023. 22

[36] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L Rus, and Song Han. BEVFusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *Proceedings of the IEEE international conference on robotics and automation*, pages 2774–2781, 2023. 15

[37] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems*, 30, 2017. 16, 18, 19, 51

[38] Shangshu Yu, Cheng Wang, Chenglu Wen, Ming Cheng, Minghao Liu, Zhihong Zhang, and Xin Li. LiDAR-based localization using universal encoding and memory-aware regression. *Pattern Recognition*, 128:108685, 2022. 16, 47, 48, 55, 57

[39] Wen Li, Shangshu Yu, Cheng Wang, Guosheng Hu, Siqi Shen, and Chenglu Wen. SGLoc: Scene geometry encoding for outdoor LiDAR localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9286–9295, 2023. 16, 18

[40] Dorian Gálvez-López and Juan D Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5): 1188–1197, 2012. 17

[41] Hervé Jégou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Pérez, and Cordelia Schmid. Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9): 1704–1716, 2011. 17

[42] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016. 17

[43] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning CNN image retrieval with no human annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1655–1668, 2018. 17, 69, 77, 106

[44] Amar Alibey, Brahim Chaibdraa, and Philippe Giguere. GSV-Cities: Toward appropriate supervised visual place recognition. *Neurocomputing*, 513:194–203, 2022. 77

[45] Amar Ali-Bey, Brahim Chaib-Draa, and Philippe Giguere. MixVPR: Feature mixing for visual place recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2998–3007, 2023. 17, 18, 77

[46] Stephen Hausler, Sourav Garg, Ming Xu, Michael Milford, and Tobias Fischer. Patch-NetVLAD: Multi-scale fusion of locally-global descriptors for place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14141–14152, 2021. 18

[47] Yingfeng Cai, Junqiao Zhao, Jiafeng Cui, Fenglin Zhang, Tiantian Feng, and Chen Ye. Patch-NetVLAD+: Learned patch descriptor and weighted matching strategy for place recognition. In *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 1–8, 2022. 18

[48] Ruotong Wang, Yanqing Shen, Weiliang Zuo, Sanping Zhou, and Nanning Zheng. TransVPR: Transformer-based place recognition with multi-level attention aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13648–13657, 2022.

[49] Gabriele Berton, Carlo Masone, and Barbara Caputo. Rethinking visual geo-localization for large-scale applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4878–4888, 2022.

[50] Sijie Zhu, Linjie Yang, Chen Chen, Mubarak Shah, Xiaohui Shen, and Heng Wang. R2Former: Unified retrieval and reranking Transformer for place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19370–19380, 2023. 18

[51] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017. 18, 19

[52] Wenxiao Zhang and Chunxia Xiao. PCAN: 3D attention map learning using contextual information for point cloud based retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12436–12445, 2019. 19, 75

[53] Juan Du, Rui Wang, and Daniel Cremers. DH3D: Deep hierarchical 3D descriptors for robust large-scale 6DoF relocalization. In *Proceedings of the European Conference on Computer Vision*, pages 744–762, 2020.

[54] Zhe Liu, Shunbo Zhou, Chuanzhe Suo, Peng Yin, Wen Chen, Hesheng Wang, Haoang Li, and Yun-Hui Liu. LPD-net: 3D point cloud learning for large-scale place recognition and environment analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2831–2840, 2019. 75

[55] Le Hui, Mingmei Cheng, Jin Xie, Jian Yang, and Ming-Ming Cheng. Efficient 3D point cloud feature learning for large-scale place recognition. *IEEE Transactions on Image Processing*, 31:1258–1270, 2022. 19, 75

[56] Jacek Komorowski. Minkloc3D: Point cloud based large-scale place recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1790–1799, 2021. 19, 75, 77

[57] Jacek Komorowski. Improving point cloud based place recognition with ranking-based loss and large batch training. In *Proceedings of the International Conference on Pattern Recognition*, pages 3699–3705, 2022. 75, 77

[58] Yan Xia, Yusheng Xu, Shuang Li, Rui Wang, Juan Du, Daniel Cremers, and Uwe Stilla. SOE-Net: A self-attention and orientation encoding network for point cloud based place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11348–11357, 2021. 19, 75

[59] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D spatio-temporal ConvNets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3075–3084, 2019. 19

[60] Peng Yin, Lingyun Xu, Ziyue Feng, Anton Egorov, and Bing Li. PSE-Match: A viewpoint-free place recognition method with parallel semantic embedding.

IEEE Transactions on Intelligent Transportation Systems, 23(8):11249–11260, 2021. 19

[61] Kamil Żywanowski, Adam Banaszczyk, Michał R Nowicki, and Jacek Komorowski. MinkLoc3D-SI: 3D LiDAR place recognition with sparse convolutions, spherical coordinates, and intensity. *IEEE Robotics and Automation Letters*, 7(2):1079–1086, 2021.

[62] Junyi Ma, Jun Zhang, Jintao Xu, Rui Ai, Weihao Gu, and Xieyuanli Chen. OverlapTransformer: An efficient and yaw-angle-invariant Transformer network for LiDAR-based place recognition. *IEEE Robotics and Automation Letters*, 7(3):6958–6965, 2022. 19

[63] Lun Luo, Shuhang Zheng, Yixuan Li, Yongzhi Fan, Beinan Yu, Si-Yuan Cao, Junwei Li, and Hui-Liang Shen. BEVPlace: Learning LiDAR-based place recognition using bird's eye view images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8700–8709, 2023. 19

[64] Yuheng Lu, Fan Yang, Fangping Chen, and Don Xie. PIC-Net: Point cloud and image collaboration network for large-scale place recognition. *arXiv preprint arXiv:2008.00658*, 2020. 20, 75

[65] Amadeus Oertel, Titus Cieslewski, and Davide Scaramuzza. Augmenting visual place recognition with structural cues. *IEEE Robotics and Automation Letters*, 5(4):5534–5541, 2020. 73, 75

[66] Haowen Lai, Peng Yin, and Sebastian Scherer. AdaFusion: Visual-LiDAR fusion with adaptive weights for place recognition. *IEEE Robotics and Automation Letters*, 7(4):12038–12045, 2022. 20, 21, 63, 64, 66, 73, 75, 76, 77, 96, 113

[67] Zijie Zhou, Jingyi Xu, Guangming Xiong, and Junyi Ma. LCPR: A multi-scale attention-based LiDAR-camera fusion network for place recognition. *IEEE Robotics and Automation Letters*, 2024. 20, 21, 76, 77, 103, 105, 113

[68] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 20

[69] Peng Gao, Jing Liang, Yu Shen, Sanghyun Son, and Ming C Lin. Visual, spatial, geometric-preserved place recognition for cross-view and cross-modal collaborative perception. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 11079–11086, 2023. 21, 103, 105, 113

[70] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 21, 87, 97, 98

[71] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 3967–3976, 2019. 22, 88, 94, 97, 98

[72] Mingi Ji, Byeongho Heo, and Sungrae Park. Show, attend and distill: Knowledge distillation via attention-based feature matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7945–7952, 2021. 22, 97, 98

[73] Ying Jin, Jiaqi Wang, and Dahua Lin. Multi-level logit distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24276–24285, 2023. 22, 97, 98

[74] Guohao Peng, Yifeng Huang, Heshan Li, Zhenyu Wu, and Danwei Wang. LSDNet: A lightweight self-attentional distillation network for visual place recognition. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 6608–6613, 2022. 22, 97, 98

[75] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 31, 2018. 23, 73

[76] Hanshu Yan, Jiawei Du, Vincent YF Tan, and Jiashi Feng. On robustness of neural ordinary differential equations. *arXiv preprint arXiv:1910.05513*, 2019. 23

[77] Qiyu Kang, Yang Song, Qinxu Ding, and Wee Peng Tay. Stable neural ode with Lyapunov-stable equilibrium points for defending against adversarial attacks. *Advances in Neural Information Processing Systems*, 34:14925–14937, 2021.

[78] Yang Song, Qiyu Kang, Sijie Wang, Kai Zhao, and Wee Peng Tay. On the robustness of graph neural diffusion to topology perturbations. *Advances in Neural Information Processing Systems*, 35:6384–6396, 2022. 24, 35

[79] Kai Zhao, Qiyu Kang, Yang Song, Rui She, Sijie Wang, and Wee Peng Tay. Adversarial robustness in graph neural networks: A Hamiltonian approach. *Advances in Neural Information Processing Systems*, 36, 2024.

[80] Kai Zhao, Qiyu Kang, Yang Song, Rui She, Sijie Wang, and Wee Peng Tay. Graph neural convection-diffusion with heterophily. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 4656–4664, 2023.

[81] Qiyu Kang, Kai Zhao, Yang Song, Yihang Xie, Yanan Zhao, Sijie Wang, Rui She, and Wee Peng Tay. Coupling graph neural networks with fractional order continuous dynamics: A Robustness study. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 13049–13058, 2024. 23

[82] Edward L Ince. *Ordinary differential equations*. Courier Corporation, 1956. 23

[83] Benjamin Paul Chamberlain, James Rowbottom, Maria Goronova, Stefan Webb, Emanuele Rossi, and Michael M Bronstein. GRAND: Graph neural diffusion. In *Proceedings of the International Conference on Machine Learning*, 2021. 24

[84] Benjamin Paul Chamberlain, James Rowbottom, Davide Eynard, Francesco Di Giovanni, Dong Xiaowen, and Michael M Bronstein. Beltrami flow and neural diffusion on graphs. In *Advances in Neural Information Processing Systems*, 2021. 24

[85] Rui She, Qiyu Kang, Sijie Wang, Wee Peng Tay, Yong Liang Guan, Diego Navarro Navarro, and Andreas Hartmannsgruber. Image patch-matching with graph-based learning in street scenes. *IEEE Transactions on Image Processing*, 32:3465–3480, 2023. 24

[86] Rui She, Qiyu Kang, Sijie Wang, Yuan-Rui Yang, Kai Zhao, Yang Song, and Wee Peng Tay. Robustmat: Neural diffusion for street landmark patch matching under challenging environments. *IEEE Transactions on Image Processing*, 32:5550–5563, 2023.

[87] Rui She, Qiyu Kang, Sijie Wang, Kai Zhao, Yang Song, Yi Xu, Tianyu Geng, Wee Peng Tay, Diego Navarro Navarro, and Andreas Hartmannsgruber. Robust graph neural diffusion for image matching. In *Proceedings of the IEEE International Conference on Image Processing*, pages 311–315, 2023. 24

[88] Rui She, Qiyu Kang, Sijie Wang, Wee Peng Tay, Kai Zhao, Yang Song, Tianyu Geng, Yi Xu, Diego Navarro Navarro, and Andreas Hartmannsgruber. PointDifformer: Robust point cloud registration with neural diffusion and Transformer. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–15, 2024. 24

[89] Rui She, Sijie Wang, Qiyu Kang, Kai Zhao, Yang Song, Wee Peng Tay, Tianyu Geng, and Xingchao Jian. PosDiffNet: Positional neural diffusion for point cloud registration in a large field of view with perturbations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 231–239, 2024. 24

[90] John M Lee and John M Lee. *Smooth manifolds*. Springer, 2012. 25, 107, 108

[91] Qi Liu, Maximilian Nickel, and Douwe Kiela. Hyperbolic graph neural networks. *Advances in Neural Information Processing Systems*, 32, 2019. 25, 26, 53

[92] Mina Ghadimi Atigh, Julian Schoep, Erman Acar, Nanne Van Noord, and Pascal Mettes. Hyperbolic image segmentation. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 4453–4462, 2022. 26, 53, 56

[93] Antonio Montanaro, Diego Valsesia, and Enrico Magli. Rethinking the compositionality of point clouds through regularization in the hyperbolic space. *Advances in Neural Information Processing Systems*, 35:33741–33753, 2022. 25, 26, 53

[94] Rik Sarkar. Low distortion delaunay embedding of trees in hyperbolic plane. In *Proceedings of the International Symposium on Graph Drawing*, pages 355–366, 2011. 26

[95] Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. *Advances in Neural Information Processing Systems*, 31, 2018. 26

[96] Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. *Advances in Neural Information Processing Systems*, 30, 2017. 26

[97] Maximillian Nickel and Douwe Kiela. Learning continuous hierarchies in the Lorentz model of hyperbolic geometry. In *Proceedings of the International Conference on Machine Learning*, pages 3779–3788, 2018. 26

[98] Shichao Zhu, Shirui Pan, Chuan Zhou, Jia Wu, Yanan Cao, and Bin Wang. Graph geometry interaction learning. *Advances in Neural Information Processing Systems*, 33:7548–7558, 2020. 26

[99] Aleksandr Ermolov, Leyla Mirvakhabova, Valentin Khrulkov, Nicu Sebe, and Ivan Oseledets. Hyperbolic vision Transformers: Combining improvements in metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7409–7419, 2022. 26

[100] Fei Xue, Xin Wu, Shaojun Cai, and Junqiu Wang. Learning multi-view camera relocalization with graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11372–11381, 2020. 30, 41, 42

[101] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The Oxford RobotCar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017. 39, 73, 96

[102] Patrick Wenzel, Rui Wang, Nan Yang, Qing Cheng, Qadeer Khan, Lukas von Stumberg, Niclas Zeller, and Daniel Cremers. 4Seasons: A cross-season dataset for multi-weather SLAM in autonomous driving. In *Pattern Recognition*, pages 404–417, 2021. 39

[103] Fei Xue, Xin Wang, Zike Yan, Qiuyuan Wang, Junqiu Wang, and Hongbin Zha. Local supports global: Deep camera relocalization with sequence enhancement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2841–2850, 2019. 41

[104] Yoli Shavit and Ron Ferens. Do we really need scene-specific pose encoders? In *Proceedings of the International Conference on Pattern Recognition*, pages 3186–3192, 2021. 42

[105] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*, pages 8748–8763, 2021. 50, 54

[106] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3523–3532, 2019. 57, 61

[107] Florian Walch, Caner Hazirbas, Laura Leal-Taixe, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based localization using lstms for structured feature correlation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 627–637, 2017. 57

[108] Yoli Shavit, Ron Ferens, and Yosi Keller. Learning multi-scene absolute pose regression with Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2733–2742, 2021. 57

[109] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 56, 73

[110] Dan Barnes, Matthew Gadd, Paul Murcutt, Paul Newman, and Ingmar Posner. The Oxford Radar RobotCar dataset: A Radar extension to the Oxford RobotCar dataset. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 6433–6438, 2020. 57

[111] Xin Lai, Yukang Chen, Fanbin Lu, Jianhui Liu, and Jiaya Jia. Spherical Transformer for LiDAR-based 3D recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17545–17555, 2023. 68, 71, 73

[112] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *Proceedings of the International Conference on Learning Representations*, pages 84–92, 2015. 72, 111

[113] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 73

[114] Keenan Burnett, David J Yoon, Yuchen Wu, Andrew Z Li, Haowei Zhang, Shichen Lu, Jingxing Qian, Wei-Kang Tseng, Andrew Lambert, Keith YK Leung, et al. Boreas: A multi-season autonomous driving dataset. *The International Journal of Robotics Research*, 42(1-2):33–42, 2023. 74, 96

[115] Lineng Chen, Huan Wang, Hui Kong, Wankou Yang, and Mingwu Ren. PTC-Net: Point-wise Transformer with sparse convolution network for place recognition. *IEEE Robotics and Automation Letters*, 2023. 75, 77

[116] Yiyuan Pan, Xuecheng Xu, Weijie Li, Yunxiang Cui, Yue Wang, and Rong Xiong. CORAL: Colored structural representation for bi-modal place recognition. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2084–2091, 2021. 75

[117] Qilong Zhang and Robert Pless. Extrinsic calibration of a camera and laser range finder. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 2301–2306, 2004. 83

[118] Lipu Zhou, Zimo Li, and Michael Kaess. Automatic extrinsic calibration of a camera and a 3D LiDAR using line and plane correspondences. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5562–5569, 2018.

[119] Nick Schneider, Florian Piewak, Christoph Stiller, and Uwe Franke. RegNet: Multimodal sensor registration using deep neural networks. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 1803–1810, 2017.

[120] Ganesh Iyer, R Karnik Ram, J Krishna Murthy, and K Madhava Krishna. CalibNet: Geometrically supervised extrinsic calibration using 3D spatial Transformer networks. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1110–1117, 2018.

[121] Xudong Lv, Boya Wang, Ziwen Dou, Dong Ye, and Shuo Wang. LCCNet: LiDAR and camera self-calibration using cost volume network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2894–2901, 2021. 83

[122] Zeyu Wang, Dingwen Li, Chenxu Luo, Cihang Xie, and Xiaodong Yang. Distill-BEV: Boosting multi-camera 3D object detection with cross-modal knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8637–8646, 2023. 87

[123] Shengchao Zhou, Weizhou Liu, Chen Hu, Shuchang Zhou, and Chao Ma. UniDistill: A Universal cross-modality knowledge distillation framework for 3D Object detection in bird's-eye view. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5116–5125, 2023. 87, 94

[124] Hui Wu, Min Wang, Wengang Zhou, Houqiang Li, and Qi Tian. Contextual similarity distillation for asymmetric image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9489–9498, 2022. 92, 97, 98

[125] Yuenan Hou, Xinge Zhu, Yuexin Ma, Chen Change Loy, and Yikang Li. Point-to-voxel knowledge distillation for LiDAR semantic segmentation. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 8479–8488, 2022. 94

[126] Le Hui, Mingmei Cheng, Jin Xie, Jian Yang, and Ming-Ming Cheng. Efficient 3D point cloud feature learning for large-scale place recognition. *IEEE Transactions on Image Processing*, 31:1258–1270, 2022. 97, 98

[127] Sijie Zhu, Taojiannan Yang, and Chen Chen. VIGOR: Cross-view image geo-localization beyond one-to-one retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3640–3649, 2021. 104

[128] Sijie Zhu, Mubarak Shah, and Chen Chen. TransGeo: Transformer is all you need for cross-view image geo-localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1162–1171, 2022. 105

[129] Yingying Zhu, Hongji Yang, Yuxin Lu, and Qiang Huang. Simple, effective and general: A new backbone for cross-view image geo-localization. *arXiv preprint arXiv:2302.01572*, 2023.

[130] Maxim Shugaev, Ilya Semenov, Kyle Ashley, Michael Klaczynski, Naresh Cuntoor, Mun Wai Lee, and Nathan Jacobs. ArcGeo: Localizing limited field-of-view images using cross-view matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 209–218, 2024. 104

[131] Yanhao Zhang, Yujiao Shi, Shan Wang, Ankit Vora, Akhil Perincherry, Yongbo Chen, and Hongdong Li. Increasing SLAM pose accuracy by ground-to-satellite image registration. *arXiv preprint arXiv:2404.09169*, 2024. 104

[132] Shan Wang, Chuong Nguyen, Jiawei Liu, Yanhao Zhang, Sundaram Muthu, Fahira Afzal Maken, Kaihao Zhang, and Hongdong Li. View from above: Orthogonal-view aware cross-view localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14843–14852, 2024. 104

[133] Guopeng Li, Ming Qian, and Gui-Song Xia. Unleashing unlabeled data: A paradigm for cross-view geo-localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16719–16729, 2024. 105

[134] Yiyi Liao, Jun Xie, and Andreas Geiger. KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2D and 3D. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3292–3310, 2022. 112

[135] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020. 112

[136] Chengfu Shu and Yutao Luo. A hierarchical and multi-modal framework for place recognition with a learnable metric. *IEEE Transactions on Intelligent Vehicles*, 2024. 113

[137] Alexander Melekhin, Dmitry Yudin, Ilia Petryashin, and Vitaly Bezuglyj. MSSPlace: Multi-sensor place recognition with visual and text semantics. *arXiv preprint arXiv:2407.15663*, 2024. 113

[138] Jingyi Xu, Junyi Ma, Qi Wu, Zijie Zhou, Yue Wang, Xieyuanli Chen, and Ling Pei. Explicit interaction for fusion-based place recognition. *arXiv preprint arXiv:2402.17264*, 2024. 113

[139] Paul-Edouard Sarlin, Eduard Trulls, Marc Pollefeys, Jan Hosang, and Simon Lynen. SNAP: self-supervised neural maps for visual positioning and semantic understanding. *arXiv preprint arXiv:2306.05407*, 2023. 113

[140] Hang Wu, Zhenghao Zhang, Siyuan Lin, Xiangru Mu, Qiang Zhao, Ming Yang, and Tong Qin. MapLocNet: Coarse-to-fine feature registration for visual re-localization in vavigation maps. *arXiv preprint arXiv:2407.08561*, 2024. 113

[141] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 114